



TREBALL FINAL DE GRAU



Estudiant: Marc Perez Arnaiz

Titulació: Grau en Enginyeria Informàtica

Títol de Treball Final de Grau: **Creació aplicació multi-plataforma**

Director/a: **Jordi Agost i Montserrat Sendín**

Presentació

Mes: Juliol

Any: 2021

CONTENIDO

1. Introducció	6
1.1. Context.....	6
1.2. Motivació	6
1.3. Objectius	7
1.3.1. Objectius funcionals.....	7
1.3.2. Objectius no funcionals	8
1.4. Resum del treball	8
1.5. Estructura del document.....	9
2. Tecnologia	10
2.1. Desenvolupament mòbil	10
2.1.1. Aplicacions natives	10
2.1.2. Aplicacions web mòbils	15
2.1.3. Aplicacions híbrides	17
3. Descripció de la tecnologia utilitzada.....	20
3.1. Introducció.....	20
3.2. Història	20
3.3. Disseny – Arquitectura	21
3.4. Widgets	22
3.5. Aplicacions	24
3.6. Casos d'èxit.....	25
3.7. IDE	26
4. Planificació del projecte	29
5. Descripció del sistema existent.....	31
5.1. Estructura actual	31
5.2. Estructura de la base de dades	31
5.2.1. Taula Users	32

5.2.2. Taula Boards	33
5.2.3. Taula Contacts.....	33
5.2.4. Taula Notifications	33
5.3. Aplicació mòbil actual	34
5.4. Model de negoci	35
6. Eines de desenvolupament.....	36
7. Descripció de la funcionalitat i anàlisis	37
7.1. Requisits funcionals.....	37
7.2. Requisits no funcionals.....	38
7.3. Diagrama casos d'us	39
7.4. Descripció diagrama casos d'us	40
7.4.1. Registrar	40
7.4.2. Login.....	40
7.4.3. Modificar perfil d'usuari	41
7.4.4. Crear taulell	41
7.4.5. Modificar taulell	42
7.4.6. Cercar usuaris.....	42
7.4.7. Enviar/Aceptar amistats	43
7.4.8. Xatejar	43
7.4.9. Tancar sessió	44
7.4.10. Valorar App	45
8. Disseny de l'aplicació.....	46
8.1. Mapa de navegació entre pantalles	46
8.2. Estil de navegació	47
8.3. Disseny de pantalles	48
8.3.1. Login.....	48
8.3.2. Registre	49
8.3.3. Pantalla d'inici	50

8.3.4. Vista perfil d'usuari.....	51
8.3.5. Taulells	52
8.3.6. Notificacions	53
8.3.7. Xats	54
8.3.1. Contactes.....	55
9. Implementació	56
9.1. Seguretat.....	56
9.2. Connexió amb la Base de Dades	58
9.3. Internalització	61
9.4. Barra de navegació	63
9.5. Missatgeria.....	66
10. Conclusions	68
11. Treballs futurs	69
12. Agraïments	70
13. Webgrafia	71

ÍNDEX D'IMATGES

Il·lustració 1 - Exemple programació Swift	12
Il·lustració 2 - Exemple programació Java	13
Il·lustració 3 - Exemple programació Kotlin	13
Il·lustració 4 - Arquitectura de Flutter	21
Il·lustració 5 - Exemple widget Text	22
Il·lustració 6 - Exemple widget Row	22
Il·lustració 7 - Exemple widget Column	23
Il·lustració 8 - Exemple widget Stack	23
Il·lustració 9 - Exemple widget Container	24
Il·lustració 10 - Estructura actual el sistema	31
Il·lustració 11 - Estructura UML de la Base de dades	32
Il·lustració 12 - Pantalles de l'aplicació	34
Il·lustració 13 - Pantalles de l'aplicació 2.....	34
Il·lustració 14 - Diagrama casos d'us	39
Il·lustració 15 - Mapa de navegació.....	46
Il·lustració 16 - Barra de navegació	47
Il·lustració 17 - AppBar	47
Il·lustració 18 - pantalla login	48
Il·lustració 19 - Pantalles de registre	49
Il·lustració 20 - Pantalla d'inici.....	50
Il·lustració 21 - Perfil d'usuari.....	51
Il·lustració 22 - Pantalla taulells.....	52
Il·lustració 23 - Pantalla notificacions	53
Il·lustració 24 - Pantalla xats	54
Il·lustració 25 - Pantalla contactes	55
Il·lustració 26 - Estructura keychain services	56
Il·lustració 27 - Estructura del sistema de connexió	59
Il·lustració 28 - Exemple de codi per a connexió php.....	59
Il·lustració 29 - Exemple fitxer php	60
Il·lustració 30 - Exemple fitxers internalització necessaris.....	61
Il·lustració 31 - Exemple internalització del text amb espanyol.....	62
Il·lustració 32 - Exemple internalització del text amb angles	62

Il·lustració 33 - Tipus de dissenys de la barra de navegació	63
Il·lustració 34 - Exemple codi creació barra de navegació	64
Il·lustració 35 - Exemple codi configuració barra de navegació	64
Il·lustració 36 - Exemple codi configuració barra de navegació 2	65
Il·lustració 37 - Exemple codi navegació a una altra pantalla	66
Il·lustració 38 - Estructura Firebase	66
Il·lustració 39 - Estructura Firebase 2	67

1. INTRODUCCIÓ

1.1. Context

Al cicle superior de desenvolupament multi-plataforma vaig tindre un primer contacte amb el desenvolupament mòbil y es on hem vaig donar conté que era el que realment a mi m'agradava, ja que el tema de programació és on més em sento còmode i desenvolupar una aplicació per a un telèfon mòbil es basa en programació amb un resultat visual molt gratificant. Ja que com va evolucionant tot, avui en dia tothom ja té un Smarthphone i crear una aplicació que potser utilitza per molta gent és una cosa que realment motiva.

El darrere any en l'assignatura de Plataformes en Xarxes va arribar el professor Jordi Agost, el qual hi havia set professor meu en el col·legi des de primària fins a l'ESO. Ell em va comentar si tenia alguna idea sobre que faria el treball de fi de grau, i en aquell moment tenia una cosa pensant, però no era gens clar. Em va comentar que volia fer una aplicació mòbil i al comentar-me aquest projecte no vaig poder dir que no, com que era el que volia fer en un principi i ja teníem un feedback de com treballàvem els dos.

1.2. Motivació

Com ja he comentat en el context anteriorment, el desenvolupament de una aplicació mòbil era el que volia fer principalment. Però l'aplicació no era una cosa tan senzilla, volia anar un pas més enllà, volia crear una aplicació amb un kit de desenvolupament software que està creixent molt aquests últims anys, el qual és anomenat Flutter¹. Un cop vaig investigar sobre Flutter i sobre els projectes de futur que tenen, vaig veure una oportunitat molt bona per aprendre a utilitzar aquest kit. Gràcies a què sempre m'ha cridat molt l'atenció, tant crear aplicacions per a Android² com per a iOS³, i amb Flutter amb un sol codi es pot utilitzar en els dos sistemes operatius.

¹ Flutter: <https://flutter.dev/>

² Android: <https://www.android.com/>

³ iOS: <https://www.apple.com/es/ios/>

També em va comentar que per fer la connexió amb la base de dades tindríem que utilitzar PHP⁴, un altre llenguatge que vaig tocar una mica al cicle però no amb profunditat. Per tant, vaig veure una oportunitat bona, de aprendre un kit nou que em sembla que serà el futur en tema de desenvolupament d'aplicacions, ja siguin mòbils o d'escriptori, i també aprendre una mica més sobre PHP.

A priori pel que em va dir, seria bastant complicat de començar, ja que havia d'aprendre des de zero molts aspectes, però això es una altra cosa que em va motivar bastant a l'hora d'acceptar aquest projecte.

1.3. Objectius

Els objectius que ens hem ficat són uns objectius curts amb la finalitat de complir l'objectiu principal que es el funcionament de l'aplicació correctament. Els objectius els hem catalogat en dos tipus que s'explicaran a continuació.

1.3.1. Objectius funcionals

Aquests objectius son de caràcter funcional, bàsicament son objectius sobre el funcionament de l'aplicació. A continuació els enumerem.

- 1- Registre d'un usuari
- 2- Creació d'un perfil d'usuari
- 3- Creació de taulells per a cada usuari
- 4- Gestió de taulells
- 5- Recerca d'usuaris registrats en l'aplicació
- 6- Notificacions
- 7- Afegir amistats
- 8- Sistema de missatgeria entre amistats
- 9- Interface senzilla
- 10- Corba d'aprenentatge rapida

⁴ PHP: <https://www.php.net/>

1.3.2. Objectius no funcionals

També ens vam ficar uns objectius que no eren directament aspectes de funcionalitat amb l'aplicació, sinó més tema personal sobre les tecnologies utilitzades, el llenguatge, etc, com es pot veure a continuació.

- 1- Aprendre el llenguatge de programació Flutter
- 2- Adquirir nous coneixements de PHP
- 3- Crear una base de dades sòlida, per tal de pugui mantenir un gran número de peticions i dades
- 4- Ampliar coneixements en gestió de dades
- 5- Aprendre a gestionar un projecte des de zero

Dins de cada objectiu anteriorment també hi ha petits objectius com ja serien objectius de tipus visual, ja que les tecnologies utilitzades en el projecte han set tecnologies que s'han tingut que anant aprenent a poc a poc mentre es desenvolupava el projecte.

Tot això, una vegada assolits tots aquests objectius, haurem complert l'objectiu principal, que es el correcte funcionament de l'aplicació explicada en el punt anterior.

1.4. Resum del treball

El treball es basa en una aplicació multi-plataforma, que es pot utilitzar tant en sistemes operatius Android com iOS. Aquesta aplicació tindria uns aspectes similars a una xarxa social i una aplicació de recerca d'usuaris per a la contractació, com podria ser LinkedIn⁵.

L'aplicació guarda els perfils d'usuaris com una xarxa social bàsica, aquest usuari tindran amistats amb altres usuaris i els podrà agrupar en taulells. Aquests taulells serveixen per tindre un control que coneixes a la persona o també per orientar a altres usuaris, en cas que el taulell sigui públic, que coneixes a aquella persona. Això ajudarà a molta gent a l'hora de buscar per exemple un perfil en concret com seria programadors.

⁵ LinkedIn: <https://es.linkedin.com/>

1.5. Estructura del document

En aquest apartat s'explica la estructura del document que hem seguit per tal de desenvolupar el projecte explicat anteriorment.

Primer de tot s'explica la **"Tecnologia"**, que tracta sobre els tipus de desenvolupament de aplicacions que hi ha.

Seguidament, podem trobar el apartat de **"Tecnologia utilitzada"**, en aquest apartat ja entrem una mica més en detall sobre la tecnologia que hem utilitzat en el projecte.

Un cop acabat tot el tema que el podríem catalogar com informatiu, entrem en el apartat de **"Planificació del projecte"**, on s'explica quina gestió hem utilitzats i els passos que hem seguit per tal de desenvolupar l'aplicació amb un cert control.

Seguidament s'explica una **"Descripció del sistema existent"**, i les **"Eines de desenvolupament"** que han set necessàries per tal de desenvolupar el projecte.

Per últim, es fa una **"Descripció de les funcionalitats i anàlisis"**, explicant també el **"Disseny de l'aplicació"** i amb la **"Implementació"** que s'ha tingut que fer per tal desenvolupar-ho tot amb èxit.

2. TECNOLOGIA

En aquest apartat de Tecnologia el que podrem veure és una explicació sobre els tipus de desenvolupaments d'aplicacions mòbils que s'utilitzen avui en dia i les eines de desenvolupament més utilitzades per a cada un dels tipus juntament amb les aplicacions més famoses de cada tipus.

2.1. Desenvolupament mòbil

El desenvolupament d'aplicacions mòbils és el procés de creació d'aplicacions de programari que s'executen en un dispositiu mòbil, i una aplicació mòbil típica utilitza una connexió de xarxa per a treballar amb recursos informàtics remots. En el desenvolupament d'aplicacions mòbils hi ha dues categories molt significatives:

2.1.1. Aplicacions natives

Aquestes aplicacions són desenvolupades específicament per a cada sistema operatiu, iOS, Android o Windows Phone, adaptant-se a cada un del llenguatge amb el qual es desenvolupa.

- Avantatges de fer aplicacions natives
 - **Aplicacions molt ràpides:** Això és pel fet que en crear unes aplicacions per a un sistema operatiu en concret saps les limitacions i el potencial del sistema operatiu. També coneixes per a quin dispositius aniran i les prestacions dels telèfons mòbils que van dirigits.
 - **Més interactiu e intuïtiu:** Cada sistema operatiu té les parts comunes que totes aplicacions desenvolupades en aquest sistema operatiu té, per tant per als usuaris és més fàcil d'aprendre a utilitzar-la.
 - **Fàcil de distribuir:** Cada sistema operatiu té la seva plataforma de distribució d'aplicacions, en el cas de iOS seria la Apple Store⁶ i en el cas de desenvolupar-la per a Android seria Google Store⁷.

⁶ App Store: <https://www.apple.com/es/ios/app-store/>

⁷ Google Store: <https://store.google.com/es/>

- Desavantatges de fer aplicacions natives
 - **Desenvolupament car i difícil de mantenir:** La majoria de desenvolupador se especialitzen en una plataforma (Android o iOS), per tant si vols realitzar l'aplicació en les dos plataformes es necessitaran dos programadors o equips separats, el que farà augmentar el temps i el cost de la aplicació.
 - **Múltiple code-bases:** Per a abastar més mercat hauràs de desenvolupar la mateixa aplicació per als dos sistemes operatius que regnen, per tant hauràs de crear dues aplicacions iguals per als dos tipus de sistemes operatius.

Uns exemples més coneguts de aplicacions natives seria:

- **Angry Birds:**



Videojoc on el protagonista és un ocell i tens que anant superant les pantalles que el joc et va oferint.

<https://play.google.com/store/apps/details?id=com.rovio.angrybirds&hl=es>

- **Pokemon GO:**



És un videojoc de realitat augmentada basat en la localització. El joc consisteix en buscar i capturar personatges basats en la saga de Pokemon.

<https://www.pokemongo.com/es-es/>

- **Twitter:**



És una xarxa social on la gent comparteix tot tipus de informació.

<https://play.google.com/store/apps/details?id=com.twitter.android&hl=es&gl=US>

En les aplicacions natives els dos sistemes operatius que regnen avui en dia en el mercat són iOS i Android, els quals tenen el 98% de la quota de mercat mundial dels sistemes operatius mòbils. En l'any 2019 la quota de mercat de iOS era del 22,17%, mentrestant la de Android per al 2022 s'estima que serà del 87%.

Actualment iOS ha aconseguit quotes de mercat d'entre el 10% i el 20% en economies emergents com el Brasil i la Xina. A Espanya la seva quota de mercat és actualment del 18%, però el Japó i els Estats Units continuen sent els seus majors mercats, amb una quota del voltant d'un 60%, segon les dades tretes de StatCounter⁸.

- iOS



És el sistema operatiu creat per als sistemes operatius de mòbils de la marca creada per Steve Jobs: Apple. aquest sistema operatiu utilitza el llenguatge de programació Swift.

Swift és un llenguatge de programació d'open source, ja que té la seva pròpia pàgina web on hi ha molts programadors de l'empresa i altres contribuents externs que comparteixen codi, reporten bugs que van sorgint i també proporciona un fòrum per a si hi ha algun dubte o per a comentar alguna millora que es pogués fer.

Aquest llenguatge també està dissenyat per a tenir molta seguretat, pel fet que elimina les classes que no són segures, les variables sempre estan controlades perquè no es puguin fer atacs de overflow i la memòria es gestiona automàticament.

```
import SwiftUI

struct ContentView: View {
    var body: some View {
        VStack {
            Text("Hello, World!")
            Text("I'm great!")
        }
    }
}

struct ContentView_Previews: PreviewProvider {
    static var previews: some View {
        ContentView()
    }
}
```

IL·LUSTRACIÓ 1 - EXEMPLE PROGRAMACIÓ SWIFT

⁸ StateCounter: <https://statcounter.com/>

- Android



És el sistema operatiu creat per l'empresa Android Inc. que va ser recolzada i posteriorment adquirida per l'empresa Google al juliol de 2005. Aquest sistema operatiu utilitza el llenguatge anomenat Java, encara que avui en dia es comença un moviment per a canviar aquest llenguatge per un altre anomenat [Kotlin](#)^[4].

Java és un llenguatge que està orientat a objectes, que facilita molt la programació, pel fet de que s'assembla molt a la manera de programar a com pensa un ésser humà. El llenguatge és un dels quals és més fàcil d'aprendre i també aquest llenguatge ve acompanyat amb unes llibreries estàndard que t'ajuden a realitzar multitud d'operacions comunes a l'hora de programar.

```
public class HolaMundo {  
  
    public static void main(String[] args) {  
        System.out.println("Hola Mundo");  
    }  
  
}
```

IL·LUSTRACIÓ 2 - EXEMPLE PROGRAMACIÓ JAVA

Kotlin és un altre llenguatge de programació que s'està implementat per al desenvolupament d'aplicacions per a Android. Es desenvolupa per la empresa de JetBrains en una de les seves oficines en San Petersburg (Rússia) a principis del juliol del 2011.

Aquest llenguatge de programació ens ajuda a l'hora de programar, per exemple, al fet que no tinguem error que un punter ens retorni nul, per tant es fa més segur per a aquests tipus de fallades. També un punt fort que té aquest llenguatge és que és més expressiu i concís, ja que amb molt menys codi fas la mateixa funció que faries amb Java, per tant retalles molt més codi i per a aplicacions amb molt de codi és més fàcil d'entendre, ja que el codi es veu reduït i és més fàcil de localitzar les errades que puguin sorgir.

```
package hello fun main(args: Array<String>) { println("Hello World!") }
```

IL·LUSTRACIÓ 3 - EXEMPLE PROGRAMACIÓ KOTLIN

Les plataformes més conegudes per a crear aplicacions natives son:

- **Android Studio:** És un entorn de desenvolupament integrat (IDE) oficial per al desenvolupaments de aplicacions per a Android, basats en IntelliJ IDEA.
- **Xcode:** És el centre de l'experiència de desenvolupament d'Apple. Estretament integrat amb els marcs de Cocoa i Cocoa Touch. Xcode és un entorn increïble productiu per crear aplicacions per a Mac, iPhone, iPad, Apple Watch i Apple TV.

- Cross-plataform

Com hem comentat anteriorment la major desavantatjada que tenen les aplicacions natives es que es té que desenvolupar la mateixa aplicació pels diferents sistemes operatius en els que vols que es pugui utilitzar. Per tant hi ha un parell de plataformes que et permeten crear el que es anomena cross-plataform, que aprofita els beneficis de del desenvolupament multi-plataforma sense sacrificar la experiència de l'usuari o el accés a les API natives.

Les plataformes més utilitzades son:

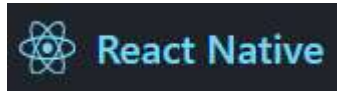
- **Xamarin:**



És una un plataforma creada per la companyia Xamarin, Inc. fundada per Miguel de Icaza i Nat Friedman el 16 de Maig de 2011, que posteriorment va ser adquirida per Microsoft el febrer de 2016. Aquesta plataforma permet al desenvolupadors crear una aplicació mòbil que funcioni en diferents plataformes, tot això utilitzant el llenguatge de programació C#. Aquesta plataforma també et proporciona un entorn de prova on pots provar l'aplicació en milers de dispositius mòbils virtuals.

<https://dotnet.microsoft.com/apps/xamarin>

- **React Native:**



Aquesta plataforma et permet crear aplicacions mòbils que no es distingirà de una aplicació creada amb Java o Objective-C. Aquesta plataforma va ser creada per Facebook, Inc. el 26 de Març del 2015, encara que la versió estable (0.63.3) va ser llançada el 29 de Setembre de 2020.

La plataforma utilitza JavaScript i React per a la creació de les aplicacions. Les aplicacions com Facebook, Tesla, i Airbnb han estat creades amb aquesta plataforma.

<https://reactnative.dev/>

2.1.2. Aplicacions web mòbils

Aquestes aplicacions són pàgines web que s'executen en dispositius mòbils. Bàsicament són aplicacions mòbils basades en web que no s'instal·len en el dispositiu en qüestió, sinó que s'executen en un servidor web. El llenguatge que utilitzen aquestes aplicacions és el mateix que s'utilitza per a pàgines web, com ara HTML, CSS, JavaScripts i JQuery.

- Els avantatges de fer aplicacions web són:
 - **No és necessari instal·lació:** Gràcies a que l'aplicació es troba en un servidor web no és necessari que l'usuari s'instal·li res en el dispositiu, per tant no ocupa espai.
 - **Sempre actualitzat:** Ja que no es necessari instal·lar, les actualitzacions es fan en el servidor i per tant l'aplicatiu sempre està actualitzat a l'última versió que l'aplicació té.
 - **Millor abast:** A l'estar en un servidor web es pot accedir des de qualsevol lloc.

- Desavantatges de fer aplicacions web
 - **No es pot accedir a funcionalitats del dispositiu:** El problema que té desenvolupar en web és que no pot accedir a funcionalitats que tenen els dispositius com seria la cambra, ubicació geogràfica, etc.
 - **Dependència de connexió a internet:** Pel fet que l'aplicació aquesta allotjada en un servidor web, el funcionament depèn de la connexió que tingui l'usuari, ja que si té una connexió dolenta l'aplicació no funcionarà com es degut i l'experiència de l'usuari serà dolenta.

Uns exemples més coneguts de aplicacions web mòbils són:

- **Shutterfly:**



És un servei en línia que permet als usuaris crear llibres de fotos, targetes personalitzades i fixes, i molt més.

<https://www.shutterfly.com/mobile/>

- **Etsy:**



És un lloc web de comerç electrònic on la gent pot comprar i vendre articles fets a mà o vintage.

<https://www.etsy.com/mobile>

Les plataformes més conegudes per a crear aplicacions híbrides son:

- **jQuery:**



Ens permet crear i dissenyar aplicacions web responsive i que seran accessibles des de qualsevol mòbil, tableta o ordinador. Aquesta plataforma utilitza el sistema de interfície de usuari basat en HTML5.

<https://jquerymobile.com/>

- **Sencha Touch:**



S'ha combinat amb Ext JS, que proporciona als desenvolupadors tot el que necessiten per crear aplicacions web que requereixen molta informació.

<https://www.sencha.com/products/touch>

2.1.3. Aplicacions híbrides

Aquest tipus d'aplicacions són aplicacions que combinen tant les aplicacions natives com les aplicacions web. Aquestes aplicacions es basen igual que les aplicacions web, en la representació d'HTML en un navegador, amb l'excepció que el navegador està integrat en l'aplicació. Igual que les aplicacions natives, les aplicacions híbrides també s'instal·len en el dispositiu i es distribueixen a través de la botiga d'aplicacions. El llenguatge d'aquestes aplicacions és el mateix que té una aplicació web, HTML, CSS, JavaScripts i JQuery.

- Avantatges de fer aplicacions híbrides
 - **Més barates que una aplicació nativa:** Pel fet que bàsicament també és una aplicació web, el preu es veu més reduït que una aplicació nativa.
 - **Només és necessari crear una aplicació:** No és necessari crear una aplicació per a cada sistema operatiu, ja que en utilitzar el llenguatge d'una aplicació web es pot instal·lar/executar en tots els sistemes operatius.
 - **No es necessita navegador:** En aquestes aplicacions no és necessari utilitzar un navegador pel fet que a ser una mescla de nadiua i web l'aplicació s'instal·la en el dispositiu.

- Desavantatges de fer aplicacions híbrides
 - **Rendiment pitjor que una aplicació nativa:** ja que no és una aplicació desenvolupada específicament per un sistema operatiu el rendiment es veu afectat.
 - **Menys interactiva que una aplicació nativa:** L'avantatge d'una aplicació nativa era que totes tenen un estàndard i són fàcils d'aprendre, a causa que aquesta no és una aplicació nativa com a tal, sinó seria més web, cada programador programa més al seu gust i per tant pot ser menys intuïtiva i interactiva que una nativa.

Les plataformes més conegudes per a crear aplicacions híbrides són:

- **Apache Cordova:**



És un entorn de desenvolupament d'aplicacions mòbils que permet als programadors de software construir aplicacions per a dispositius mòbils utilitzant HTML, CSS3 i JavaScripts.

<https://cordova.apache.org/>

- **Canvas:** Ofereix un servei per a qualsevol que tingui una pagina web a crear una aplicació mòbil per a qualsevol sistema operatiu.

Uns exemples més coneguts de aplicacions web mòbils són:

- **Marketwatch:**



Proporciona les últimes notícies borsàries, financeres i comercials. Obtingui cotitzacions borsàries, assessorament financer personal, notícies de l'empresa i més.

<https://www.marketwatch.com/site/mobile>

- **Sworkit:**



Permet crear i seguir entrenaments personalitzats via vídeos.

<https://sworkit.com/>

- **Untappd:**



És una xarxa social que permet als seus usuaris realitzar check-ins de cerveses a mesura que les consumeixen i compartir aquests check-ins i la seva localització.

<https://untappd.com/>

3. DESCRIPCIÓ DE LA TECNOLOGIA UTILITZADA

En aquest projecte la tecnologia utilitzada ha estat la anomenada Flutter. A continuació, s'explica tots els aspectes d'aquesta tecnologia amb detall.

3.1. Introducció

Flutter^[9] és una plataforma que et permet crear aplicacions cross-plataform, per tant com s'ha explicat anteriorment aquesta tecnologia cross-plataform, al compilar produeixen aplicacions completament natives sense la intervenció d'algun navegador web per a la seva execució com seria les aplicacions híbrides.

3.2. Història

Flutter va ser anunciat en el any 2005 per la companyia Dart Developer Summit amb el nom de SKY, més endavant el nom canviaria al que ara coneixem com Flutter. SKY es una plataforma on et permet crear aplicacions anomenades cross-plataform que permeten crear aplicacions completament natives per als dos sistemes operatius més utilitzat com són Android e iOS, sense la necessitat de crear dos aplicacions per a cada sistema operatiu.

La primera versió va ser una alpha (V-0.06) que va ser llançada al 2017. Més endavant Google va llançar la segona vista al públic al Setembre de 2018. En aquest últim llançament la gent ja va començar a parlar d'ella i es veia el potencial que aquesta plataforma et proporcionava. Abans del llançament de la versió 1.0 ja es van començar a anunciar que aplicacions com Alibaba, Google Ads i altres més arribarien al mercat totes elles creades amb la plataforma Flutter.

Finalment, Google el dia 5 de Desembre de 2018 va llançar la versió 1.0 de Flutter on ja era una versió estable.

Flutter utilitza un llenguatge de programació anomenat Dart. Aquest llenguatge és un llenguatge de programació orientat a objectes desenvolupat per Google i és molt fàcil d'aprendre degut a que s'assembla molt a Java, C# o JavaScript.

La primera versió que es va tindre de Dart va ser llançada al 2011 i es diu que Dart va ser creat com una alternativa per a JavaScript, encara que els desenvolupadors ho neguen.

Actualment la última versió de Flutter es la 2.2 que va ser anunciada al Google I/O Edition 2021. Aquesta versió està construït sobre la base de Flutter 2, que va estendre Flutter des de les seves arrels mòbils per a incorporar l'ús web, d'escriptori i embegut. Flutter 2.2 també ve amb la versió nova de Dart 2.13 que conté una nova funció d'àlies de tipus, que li permet crear àlies per a tipus i funcions. Els àlies de tipus permeten donar noms curts agradables a tipus llargs i complicats, i també li permet canviar el nom de les seves classes sense interrupcions.

3.3. Disseny – Arquitectura

Flutter esta dissenyat com un sistema extensible de capes. Existeix com una sèrie de biblioteques independents que depenen de la capa subjacent, però ninguna capa té accés privilegiat a la capa inferior, i cada part del nivell del marc està dissenyat per a ser opcional i reemplaçada.

En la imatge següent podem apreciar un exemple de com seria l'arquitectura que té:



IL·LUSTRACIÓ 4 - ARQUITECTURA DE FLUTTER

La primera capa seria Material i Cupertino que ens ofereix un conjunt complet d'opcions per a especificar la interfície d'usuari que utilitza la composició de la capa de widget.

En la segona capa trobem els Widgets. En aquesta capa cada objecte renderitzat en la capa de renderitzat té una classe corresponent en aquesta capa per tindre un major control, i a més a més aquesta capa et permet definir combinacions de classes que en un futur podràs reutilitzar.

La tercera capa és la capa de renderitzat, que et proporciona una abstracció per a poder treballar amb el disseny gràfic. En aquesta capa pots anar construint un arbre i els canvis que vas generant s'actualitzen automàticament.

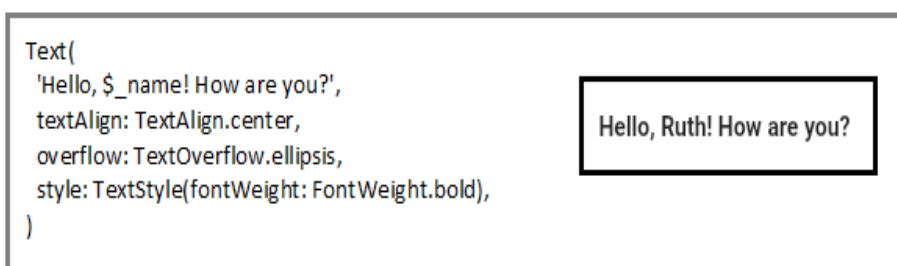
Per últim, trobem la capa Foundation. En aquesta capa trobem les classes bàsiques i serveis de blocs de construcció com són Animation, Painting i Gestures, que ens ofereixen una abstracció d'ús comú sobre la base subjacent, perquè sigui més fàcil d'utilitzar i entendre l'ús d'aquestes.

3.4. Widgets

Els anomenats widgets són, bàsicament, classes Dart utilitzades per a construir la teva interfície d'usuari de l'aplicació. Cada widget és el encarregat del seu renderitzat basant-se en la seva configuració i el seu estat intern. Flutter ens proporciona una varietat de widget basic per a la creació de la interfície d'usuari. Encara que també podem trobar widgets creats per tercers on es podem pujar i ser avaluats per els desenvolupadors i els usuaris en una web proporcionada per els creadors de Flutter anomenada <https://pub.dev/>

Els widgets més basics que et proporciona Flutter són:

- **Text:** Ens permet afegir un text amb estil dins de la aplicació. Exemple:



IL·LUSTRACIÓ 5 - EXEMPLE WIDGET TEXT

- **Row, Column:** Ens permet crear un disseny flexible en les direccions horitzontals (files) i verticals (columnes). Exemple:



IL·LUSTRACIÓ 6 - EXEMPLE WIDGET ROW

```

Column(
  mainAxisAlignment: MainAxisAlignment.spaceEvenly,
  children: [
    Image.asset('images/pic1.jpg'),
    Image.asset('images/pic2.jpg'),
    Image.asset('images/pic3.jpg'),
  ],
);
  
```



IL·LUSTRACIÓ 7 - EXEMPLE WIDGET COLUMN

- **Stack:** Ens permet widgets uns sobre els altres en ordre de pintura. Aquest widget es basa en el model de disseny de posicionament absolut del web. Exemple:

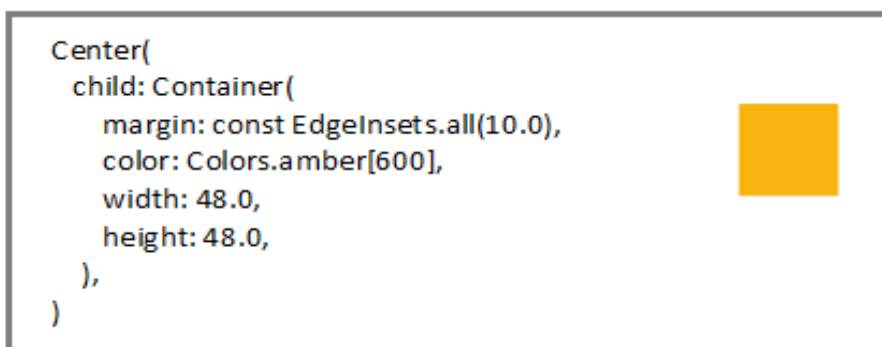
```

Stack(
  children: <Widget>[
    Container(
      width: 100,
      height: 100,
      color: Colors.red,
    ),
    Container(
      width: 90,
      height: 90,
      color: Colors.green,
    ),
    Container(
      width: 80,
      height: 80,
      color: Colors.blue,
    ),
  ],
);
  
```



IL·LUSTRACIÓ 8 - EXEMPLE WIDGET STACK

- **Container:** Ens permet crear un element visual rectangular. En aquest widget es pot decorar amb una decoració BoxDecoration, que et permet canviar el fons, ombra, etc. També un contenidor es pot transformar en un espai tridimensional mitjançant una matriu. Exemple:



IL·LUSTRACIÓ 9 - EXEMPLE WIDGET CONTAINER

Tots els widgets poden ser statless o stateful, tot depèn de si el widget pot canviar o no (per exemple, quan un usuari interacciona amb ell), llavors significa que és stateful. L'estat d'aquests widgets es guardarà en un objecte de tipus State, que separa el estat del widget de la seva aparença. Un exemple d'aquest widgets seria CheckBox, Form o TextField.

Els widgets que mai canvien d'estat son aquells anomenats widgets stateless que tindrà la subclasse de StatelessWidget. Un exemple són els widgets Icon, IconButton o Text.

3.5. Aplicacions

Quan va arribar la versió 1.5 de Flutter en la conferència Google I/O 2019 es va anunciar que Flutter passaria de ser un framework per al desenvolupaments exclusiu d'aplicacions mòbils a ser un framework multi-plataforma per a crear aplicacions que funcionin en mòbil, web, ordinadors i embeguts.

També es va anunciar Technical Preview per a Flutter i web, per a la creació d'aplicacions web i aplicacions web progressives utilitzant Flutter.

Per últim, també es va mostrar una primera versió de que seria Flutter per a Desktop, on es podrà desenvolupar aplicacions per a Windows, Linux i macOS. Per com es va comentar en la conferència, tant Flutter per a Desktop com per a web, estan en una primera versió i encara tenen un llarg camí per a ser una bona plataforma per a desenvolupar aquest tipus d'aplicacions.

Com s'ha comentant en el apartat anterior, a la conferència de Google I/O 2021 que es va realitzar el 28 de maig, ja es va anuncia una nova versió de Flutter 2.2, que incorporar l'ús web, d'escriptori i embegut. També es va ensenyar que hi ha més de 200.000 aplicacions a la Play Store desenvolupades únicament amb Flutter. Per últim, es va mostrar el numero de descarregues que tenen aquestes aplicacions com seria per exemple WeChat⁹, desenvolupada per la empresa Tencent¹⁰, que ronda el 1,2 billons d'usuaris en Android e iOS.

3.6. Casos d'èxit

A continuació es podran veure uns exemples de les aplicacions més famoses creades amb Flutter i l'opinió dels creadors d'aquestes aplicacions.

- **Alibaba:**



Alibaba és la empresa de comerç en línia més gran del món. Aquesta aplicació té més de 50 milions de descarregues i més de 10 milions de usuaris diaris. El creador Xianyu comenta que van utilitzar Flutter per a la creació de la aplicació, ja que, els estalviaria molt de temps gràcies a que sol necessiten un únic code base per a la creació de la aplicació tant per a Android com per a iOS, i degut a que tenen una desena de milions d'usuaris no podien crear una aplicació des de zero, per tant el que van fer es anar implementant funcionalitats de Flutter de forma incremental.

<https://app.alibaba.com/>

⁹ WeChat: <https://www.wechat.com/es/>

¹⁰ Tencent: <https://www.tencent.com/en-us/about.html>

- **Topline:**



Topline és una aplicació que permet als artistes enregistrar i desar les seves idees de cançons allà on siguin. El creador comenta que Flutter va millorar i va el desenvolupament de la seva aplicació. L'equip encarregat anomenat Miquido va trigar deu setmanes per a la creació de l'aplicació. També ens comenta que el que té de bo Flutter és que et proporciona una gran biblioteca de widgets que ja estan preparats per a l'ús, que són personalitzables al teu gust i que funcionen tal i com tu preveus. Per últim comentar que una funcionalitat que et proporciona Flutter, anomenada Hot Reload, els va ajudar molt als desenvolupadors a millorar ràpidament, gràcies a aquesta funcionalitat es pot trobar els errors molt més ràpid i solucionar-los en temps real.

<https://www.abbeyroad.com/apps>

3.7. IDE

Per a la creació d'una aplicació amb Flutter es pot utilitzar qualsevol tipus editor de text combinat amb les eines de línia de comandes que et proporciona el equip desenvolupador de Flutter, encara que hi ha tres editor de text recomanats pels desenvolupadors, els quals són:

- **Android Studio e IntelliJ:**

- Android Studio:



És l'entorn de desenvolupament integrat oficial per a la plataforma de Android. El autor d'aquest entorn va ser al empresa anomenada JetBrains, que és una companyia de desenvolupament de software dirigida a els desenvolupadors de software i gerents de projectes. El desenvolupador de Android Studio actualment és Google i va ser llançada el 8 de Desembre del 2014. Actualment l'última versió del software és la 4.0 que va ser publicada el 28 de Maig del 2020.

Android Studio ofereix una experiència IDE completa i integrada per a Flutter. Pel correcte funcionament de Flutter és necessari instal·lar una versió 3.0 o superior.

<https://developer.android.com/studio>

- **IntelliJ IDEA:**



És un entorn de desenvolupament integrat pel desenvolupament de programes informàtics. És desenvolupat per la companyia JetBrains, i està disponible en dues edicions: Community i Ultimate.

La versió necessàries per al correcte funcionament, tant per a l'edició Community com la Ultimate, és la versió 2017.1 o superior.

<https://www.jetbrains.com/idea/download/#section=windows>

Per a la instal·lació dels plugins de Flutter i Dart, en els dos softwares mencionats anteriorment, és necessari seguir els següents passos:

1. Iniciï Android Studio/IntelliJ IDEA.
2. Obrir les preferències de complements (Configurar > Complementos a partir de v3.6.3.0 o posterior).
3. Selecciona el complement Flutter i fes clic a Instal·lar .
4. Fes clic en Si quan se li sol·liciti que instal·li el complement Dart.
5. Fes clic a Reiniciar quan se li sol·liciti.

- **Visual Studio Code:**



És un entorn de desenvolupament integrat, creat per la companyia Microsoft i disponible per a sistemes operatius Windows, Linux i macOS, i la vegada és compatible amb múltiples llenguatges de programació, com ara C++, C#, Visual Basic .NET, F#, Java, Python, Ruby i PHP.

<https://code.visualstudio.com/>

Per a la instal·lació de dels complements de Flutter i Dart s'ha de seguir els següents passos:

1. Iniciï VS Code.
2. Invocar Vista > Paleta de comandos...
3. Escrigui "instal·lar" i seleccioni Extensions: Instal·lar extensions
4. Escrigui "flutter" en el camp de cerca d'extensions, seleccioni Flutter en la llista i faci clic a Instal·lar. Això també instal·la el complement de Dart requerit.

▪ **Emacs:**



És un editor de text amb una gran quantitat de funcions, a més a més és un editor molt popular entre el programadors i usuaris tècnics. Els autors d'aquet editor són David Moon, informàtic nord-americà i desenvolupador de Lisp (llenguatge de programació), Richard Stallman, programador nord-americà i fundador del moviment del software lliure, i per últim, Guy Lewis Steele Jr., que és un informàtic nord-americà que ha jugat un paper important en el disseny i documentació de diversos llenguatges de programació i estàndards tècnics.

<https://www.gnu.org/software/emacs/download.html>

Per a poder instal·lar Flutter i Dart en aquest editor de text l'únic que s'ha de fer es instal·lar el paquet lsp-dart.

<https://emacs-lsp.github.io/lsp-dart/>

4. PLANIFICACIÓ DEL PROJECTE

Pel desenvolupament del projecte hem utilitzat la gestió anomenada Agile. La gestió de projectes Agiles és un conjunt de metodologies per al desenvolupament, sobretot per a projectes de desenvolupament de software o bàsicament del món d'Internet, que precisa d'una rapidesa i flexibilitat en el seu procés. Això ens ha anat bé com que el projecte és un projecte de fi de grau i el temps per desenvolupar-ho es un període bastant curt, i gràcies a Agile ens facilita tindre un control millor del temps.

Les avantatges de treballar amb aquest gestió de projectes són les següents:

1. **Treball col·laboratiu.** Tots els implicats en el projecte han tingut reunions setmanals per veure els problemes que anaven sorgint i veure com enfocar el següent pas a seguir per avançar en el desenvolupament.
2. **Major control i capacitat de predicció.** L'oportunitat de revisar i adaptar el producte al llarg del procés àgil permet a tots els membres del projecte exercir un major control sobre el seu treball.

El manifest Àgil és conforma de quatre valors principals els quals són:

1. Individus i interaccions sobre processos i eines.
2. 'Programari' funcionant sobre documentació exhaustiva.
3. Col·laboració amb el client sobre negociació contractual.
4. Resposta davant el canvi sobre seguir un pla.

I també ens conforma en dotze principis. A continuació es mostrà els principis del Manifest Àgil.

5. La nostra major prioritat és satisfer al client mitjançant el lliurament primerenc i continu de 'programari' amb valor.
6. Acceptem que els requisits canviïn, fins i tot en etapes tardanes del desenvolupament. Els processos àgils aprofiten el canvi per a proporcionar avantatge competitiu al client.
7. Lliurem 'programari' funcional sovint, entre dues setmanes i dos mesos, preferentment en el període de temps més curt possible.
8. Els responsables de negoci i els desenvolupadors treballem junts de manera quotidiana durant tot el projecte.
9. Els projectes es desenvolupen entorn d'individus motivats. Cal donar-los l'entorn i el suport que necessiten, i confiar-los l'execució del treball.

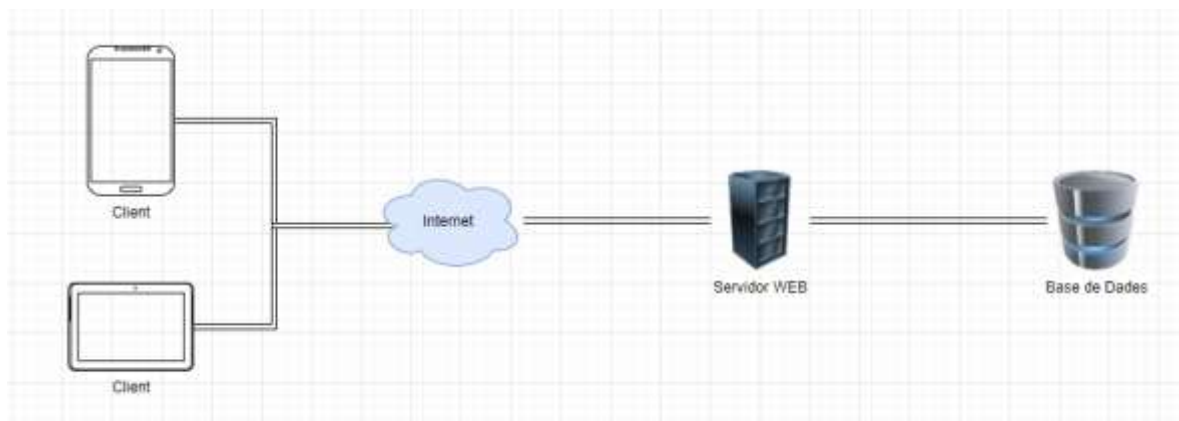
10. El mètode més eficient i efectiu de comunicar informació a l'equip de desenvolupament, i entre els membres de l'equip, és la conversa cara a cara.
11. El 'programari' funcionant és la mesura principal de progrés.
12. Els processos àgils promouen el desenvolupament sostenible. Els promotors, desenvolupadors i usuaris hem de ser capaços de mantenir un ritme constant de manera indefinida.
13. L'atenció contínua a l'excel·lència tècnica i al bon disseny millora l'agilitat.
14. La simplicitat, o l'art de maximitzar la quantitat de treball no realitzat, és essencial.
15. Les millors arquitectures, requisits i dissenys emergeixen d'equips auto-organitzats.
16. A intervals regulars l'equip reflexiona sobre com ser més efectiu per a la continuació ajustar i perfeccionar el seu comportament en conseqüència.

Aquest manifest va ser creat el 17 de Febrer del 2001 per 17 autors que es van reunir en la estació d'esquí Snowbird i van ser els primers agilistes. Tots aquests autors eren experts en programació i desenvolupament de software i van crear un document on es resumeix en quatre valors i dotze principis la filosofia Àgil.

5. DESCRIPCIÓ DEL SISTEMA EXISTENT

En aquest apartat veurem amb més detall tot el relacionat amb el projecte, tant l'infraestructura implementada per al correcte funcionament de l'API amb la base de dades, com la implementació més detallada de la base de dades.

5.1. Estructura actual

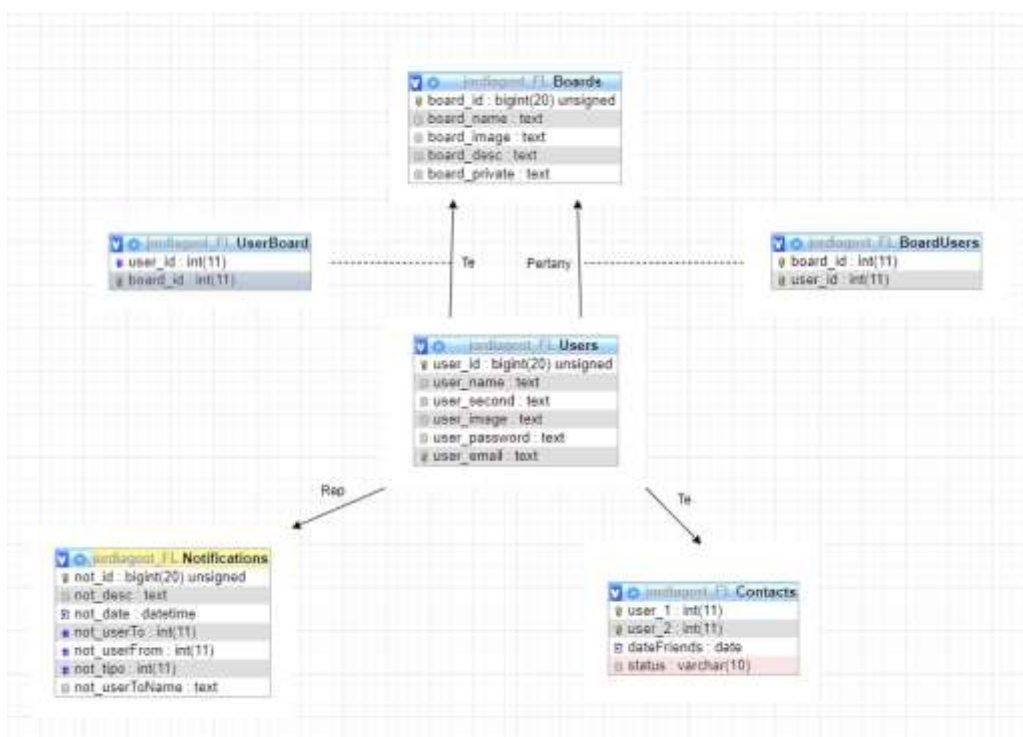


IL·LUSTRACIÓ 10 - ESTRUCTURA ACTUAL EL SISTEMA

Com es pot veure en la imatge anterior, l'estructura, bàsicament, és una estructura client/servidor típica, l'única diferència que el servidor, en aquest cas, és el que fa d'intermediari entre l'API i la base de dades a través de consultes amb PHP.

5.2. Estructura de la base de dades

En aquest apartat es pot veure el model relacional que mostra l'estructura de la base de dades actual, on totes les consultes futures que es realitzin de l'API s'obtidran d'aquest model.



IL·LUSTRACIÓ 11 - ESTRUCTURA UML DE LA BASE DE DADES

A continuació s'explica una breu explicació de la relació i del perquè aquestes taules.

5.2.1. Taula Users

En aquesta taula es guardarà tota la informació necessària per al correcte funcionament de l'aplicació de l'usuari. Com es pot veure en el diagrama anterior es guardarà informació sensible amb la contrasenya que l'usuari ha introduït en la creació del compte. Per tema de seguretat no es guarda la contrasenya en si, sinó un Hash creat per la base de dades per tal que si per algun error de seguretat les dades dels usuaris han set compromeses, no puguin obtenir la contrasenya dels usuaris.

5.2.2. Taula Boards

Com el nom de la taula indica, aquí es guardarà la informació bàsica de les taules que els usuaris van creant, com seria el nom, una petita descripció i també un camp on indicarà a l'aplicació si aquest taulell és privat o públic.

Entre la taula Users i Boards podem trobar dos relacions:

- Taula UserBoard. Relació on ens indicarà de quin usuari es cada taulell.
- Taula BoardUsers. Relació on ens mostrarà quins usuaris estan en cada taulell.

5.2.3. Taula Contacts

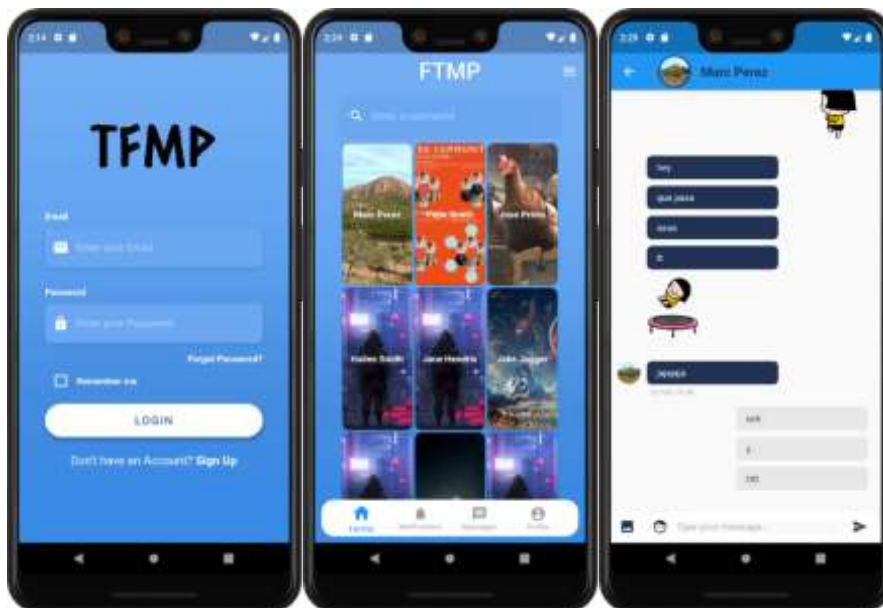
En la taula de Contacts ens mostrarà si els usuaris en qüestió són amics o han enviat petició d'amistat a un altre usuari amb la data de quan han enviat aquesta petició. Per saber si aquests usuaris són amics o li han enviat una petició, s'utilitza el camp status, el qual a l'enviar una petició s'indiciarà amb un "pending". Un cop l'usuari en qüestió accepti aquesta petició, el camp status canviarà a "friends", per saber que aquests dos usuaris ara són amics.

5.2.4. Taula Notifications

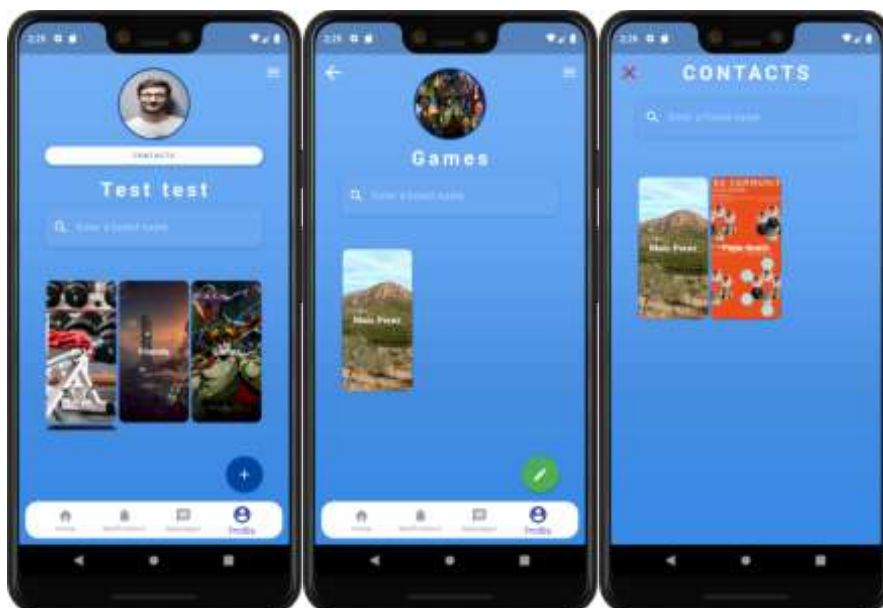
La taula notificacions registrarà tot tipus de notificacions que un usuari pot rebre. En aquest cas el tipus de la notificació pot ser de tipus 0 que significarà que has rebut una notificació d'amistat o de tipus 1, que voldrà dir que la sol·licitud d'amistat enviada a un usuari ha set acceptada.

5.3. Aplicació mòbil actual

A continuació es podran veure imatges d'algunes de les pantalles que l'aplicació té implementades. Aquestes imatges són de les pantalles en el qual l'usuari interactuarà més quan utilitzi l'aplicació.



IL·LUSTRACIÓ 12 - PANTALLES DE L'APLICACIÓ



IL·LUSTRACIÓ 13 - PANTALLES DE L'APLICACIÓ 2

5.4. Model de negoci

L'aplicació té una limitació en la creació de taulells, per tant en cas de que una persona vulgui crear més taulells, ampliar la seva llista de contactes i treure la publicitat que l'aplicació ha acceptat a mostrar, tindria que pagar mensualment 9.99€. Degut a que gràcies a la publicitat, l'aplicació podrà mantindre els serveis encara que ningun usuari estigui disposat a pagar la mensualitat per aquestes millores en l'aplicació. Aquesta opció es anomenada Freemium, ja que no es necessària per al funcionament de la aplicació correctament, l'únic que et dona és una sèrie de millores en l'aplicació.

Una altra opció de model de negoci és la opció de Subscripció com empresa, la qual et permetrà tindre un distintiu sobre els altres usuaris i la creació de molts més taulells i usuaris en els taulells. Aquesta subscripció tindrà que ser revisada per algun moderador de l'aplicació per comprovar que es tracta de una empresa.

6. EINES DE DESENVOLUPAMENT

A continuació és mostra una llista amb totes les eines que han set utilitzades en el desenvolupament de l'aplicació i una petita descripció del seu funcionament.

- **PhpmyAdmin:** És un programa de lliure distribució en PHP, creat per una comunitat sense ànim de lucre, que només treballa en el projecte per amor a l'art. És una eina molt completa que permet accedir a totes les funcions típiques de la base de dades MySQL a través d'una interfície web molt intuïtiva.
- **AndroidStudio:** És l'entorn de desenvolupament integrat (IDE) oficial per al desenvolupament d'apps per a Android i està basat en IntelliJ IDEA. A més del potent editor de codis i les eines per a desenvolupadors de IntelliJ, Android Studio ofereix fins i tot més funcions que augmenten la teva productivitat quan desenvolupes apps per a Android.
- **Atom:** És un editor de codi font de codi obert per a macOS, Linux, i Windows amb suport per a múltiples plug-in escrits en Node.js i control de versions Git integrat, desenvolupat per GitHub.
- **FileZilla Client:** És una aplicació FTP lliure i de codi obert que consta d'un client i un servidor. El que ens permetrà es connectar-nos amb el servidor i poder pujar els fitxers php, els quals l'API atacarà per fer les peticions a la base de dades.

7. DESCRIPCIÓ DE LA FUNCIONALITAT I ANÀLISIS

A continuació es mostra totes les funcionalitats que han set implementades en l'aplicació, tant funcionalitats que l'usuari executa, com funcionalitats que l'aplicació executa per tal de complir amb el correcte funcionament.

7.1. Requisits funcionals

La llista de requisits de a continuació són requisits essencials pel funcionament correctament en tema de les accions que el usuari pot efectuar en l'aplicació.

- **RF1:** Permetre crear un usuari a la bbdd per a l'aplicació
- **RF2:** Permetre iniciar sessió dels usuaris de la bbdd a l'aplicació
- **RF3:** Permetre als usuaris de la bbdd canviar la contrasenya
- **RF4:** Permetre al usuari veure perfil d'usuaris
- **RF5:** Permetre al usuari crear taulells
- **RF6:** Permetre al usuari modificar els taulells de l'usuari
- **RF7:** Permetre al usuari afegir usuaris als taulells
- **RF8:** Permetre al usuari recerca de taulells a traves del nom
- **RF9:** Permetre al usuari eliminar usuaris dels taulells
- **RF10:** Permetre al usuari visualitzar taulells públics d'altres usuaris
- **RF11:** Permetre al usuari enviar sol·licitud d'amistat
- **RF12:** Permetre al usuari recerca d'usuari a traves del nom
- **RF13:** Permetre al usuari acceptar sol·licituds d'amistat
- **RF14:** Permetre al usuari veure un llistat amb totes les amistats de l'usuari
- **RF15:** Permetre al usuari eliminar una amistat
- **RF16:** Permetre al usuari veure les notificacions de l'usuari
- **RF17:** Permetre al usuari enviar missatge a una amistat
- **RF18:** Permetre al usuari rebre un missatge d'una amistat
- **RF19:** Permetre al usuari veure tota la conversa amb una amistat

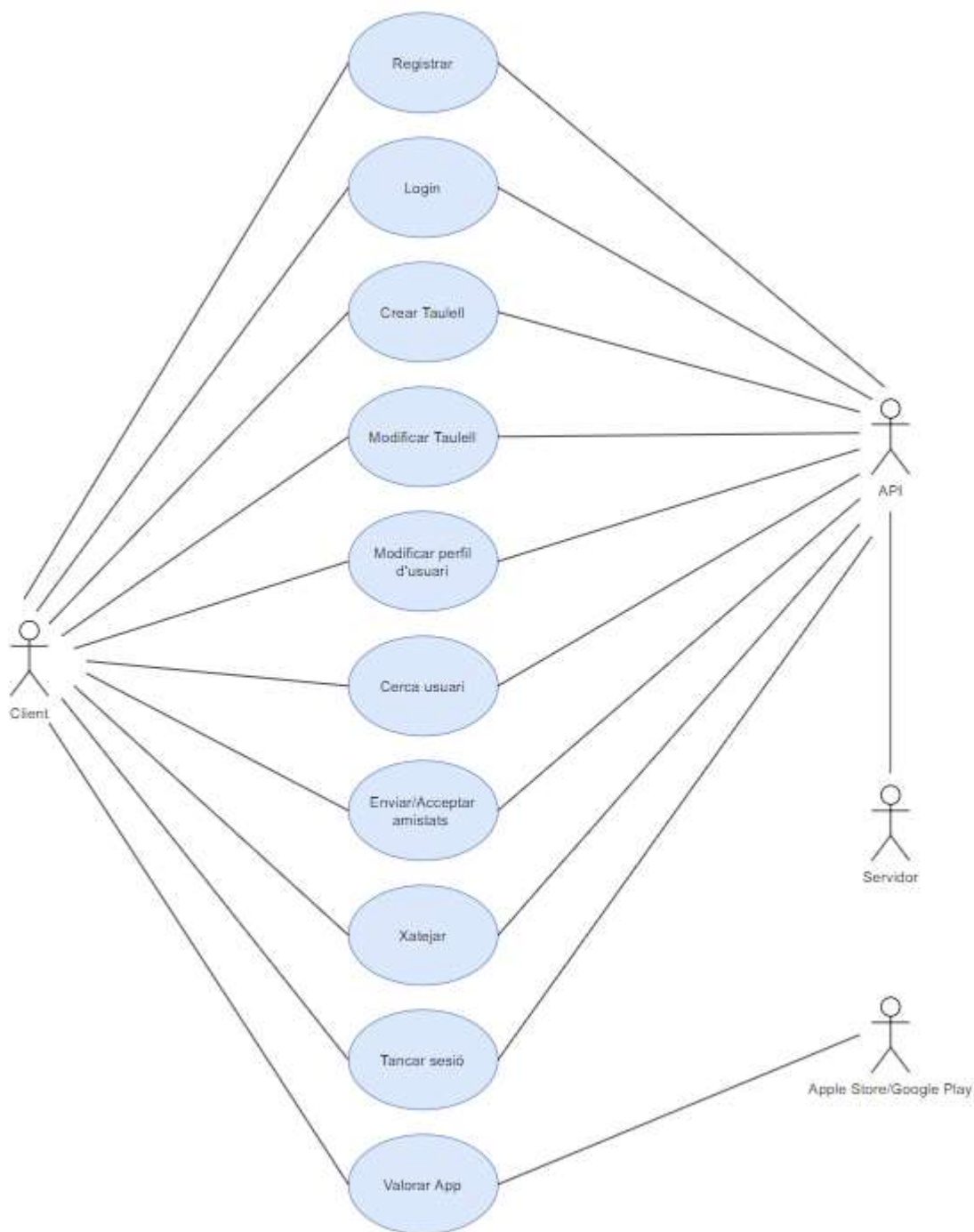
7.2. Requisits no funcionals

Aquest requisits son requisits que no influeixen directament en el funcionament de l'aplicació però són necessaris per tal de facilitar a l'usuari l'ús de l'aplicació.

- **RNF1:** Totes les funcionalitats de l'aplicació han de ser senzilles i fàcils d'aprendre. Qualsevol persona familiaritzada amb un dispositiu intel·ligent li sigui fàcil d'utilitzar
- **RNF2:** Tindre un control de les dades introduïdes per l'usuari
- **RNF3:** Missatges d'errors clars i amb colors intuïtius
- **RNF4:** Compliment amb la legislació vinent de protecció de dades
- **RNF5:** Per a l'ús de l'aplicació serà necessari tindre un terminal amb un sistema operatiu Android amb API level mínim 18 o iOS, independentment de si es tracta de una tableta tàctil o un telèfon mòbil
- **RNF6:** La mida de l'aplicació ha de ser inferior a 100MB per a poder ser pujada a la PlayStore i menys de 150MB per a l'AppStore.

7.3. Diagrama casos d'us

A continuació es mostra un diagrama on es pot veure totes els actors i casos d'us que té l'aplicació.



IL·LUSTRACIÓ 14 - DIAGRAMA CASOS D'US

7.4. Descripció diagrama casos d'us

7.4.1. Registrar

- **Propòsit:** L'usuari desitja registrar-se en l'aplicació
- **Precondició:** L'usuari no està registrat en el sistema
- **Postcondició:** Es crear un nou perfil d'usuari amb les dades introduïdes

Curs típic d'esdeveniments:	
Accions dels autors	Respostes del sistema
1. L'usuari va a la pantalla de registre	
	2. El sistema mostrarà el formulari de registre
3. L'usuari completa el formulari	
	4. El sistema comprova les dades al backend
	5. El sistema mostrarà el formulari de creació del perfil d'usuari
6. L'usuari completa el formulari	
	7. El sistema envia les dades al backend

7.4.2. Login

- **Propòsit:** L'usuari desitja iniciar sessió en l'aplicació
- **Precondició:** L'usuari està registrat en el sistema
- **Postcondició:** L'usuari entra en l'aplicació

Curs típic d'esdeveniments:	
Accions dels autors	Respostes del sistema
1. L'usuari executa l'aplicació	
	2. El sistema mostrarà el formulari de logueig
3. L'usuari completa el formulari	
	4. El sistema comprova les dades al backend

7.4.3. Modificar perfil d'usuari

- **Propòsit:** L'usuari desitja modificar el perfil d'usuari
- **Precondició:** L'usuari està registrat en el sistema
- **Postcondició:** L'usuari modifica les dades correctament del seu perfil

Curs típic d'esdeveniments:	
Accions dels autors	Respostes del sistema
1. L'usuari va a la pantalla del perfil	
	2. El sistema mostrarà el formulari amb la informació actual del perfil
3. L'usuari modifica el formulari	
	4. El sistema comprova les dades al backend
	5. El sistema et redirigeix a la pantalla del perfil

7.4.4. Crear taulell

- **Propòsit:** L'usuari desitja crear un taulell en l'aplicació
- **Precondició:** L'usuari ha fet login en l'aplicació
- **Postcondició:** L'usuari crear un taulell correctament en l'aplicació

Curs típic d'esdeveniments:	
Accions dels autors	Respostes del sistema
1. L'usuari va a la pantalla de creació de taulells	
	2. El sistema mostra el formulari
3. L'usuari completa el formulari	
	4. El sistema comprova les dades al backend

7.4.5. Modificar taulell

- **Propòsit:** L'usuari desitja modificar un taulell en l'aplicació
- **Precondició:** L'usuari ha fet login en l'aplicació i té creat el taulell
- **Postcondició:** L'usuari modifica la informació d'un taulell correctament

Curs típic d'esdeveniments:	
Accions dels autors	Respostes del sistema
1. L'usuari va a la pantalla del taulell	
	2. El sistema mostrarà el formulari amb la informació actual del taulell
3. L'usuari modifica el formulari	
	4. El sistema comprova les dades al backend
	5. El sistema et redirigeix a la pantalla del taulell

7.4.6. Cercar usuaris

- **Propòsit:** L'usuari desitja buscar un usuari en l'aplicació
- **Precondició:** L'usuari ha fet login en l'aplicació
- **Postcondició:** L'usuari troba l'usuari en l'aplicació

Curs típic d'esdeveniments:	
Accions dels autors	Respostes del sistema
1. L'usuari va a la pantalla principal	
	2. El sistema accedeix al backend i mostra tots els usuaris
3. L'usuari escriu un nom al buscador	
	4. El sistema filtre el nom i mostrarà similituds

7.4.7. Enviar/Aceptar amistats

- **Propòsit:** L'usuari desitja tindre una amistat amb un usuari de l'aplicació
- **Precondició:** L'usuari ha fet login en l'aplicació
- **Postcondició:** L'usuari ara té amistat amb l'altre usuari

Curs típic d'esdeveniments:	
Accions dels autors	Respostes del sistema
1. L'usuari 1 envia sol·licitud d'amistat a un altre usuari 2	
	2. El sistema processa la petició al backend
	3. El sistema envia una notificació al usuari 2
4. L'usuari 2 accepta la sol·licitud	
	5. El sistema processa la petició al backend
	6. El sistema envia una notificació al usuari 1

7.4.8. Xatejar

- **Propòsit:** L'usuari desitja enviar un missatge a un altre usuari
- **Precondició:** L'usuari ha fet login en l'aplicació i té amistat amb l'usuari
- **Postcondició:** L'usuari envia correctament un missatge amb l'altre usuari

Curs típic d'esdeveniments:	
Accions dels autors	Respostes del sistema
1. L'usuari va a la pantalla de xats	
2. L'usuari selecciona un usuari	
	3. El sistema accedeix al backend per obtenir tota la conversa entre els usuaris
	4. El sistema mostrarà tota la conversa
5. L'usuari escriu un missatge	
	6. El sistema mostra el missatge
	7. El sistema processa el missatge i l'envia al backend

7.4.9. Tancar sessió

- **Propòsit:** L'usuari desitja tancar la sessió
- **Precondició:** L'usuari ha fet login en l'aplicació
- **Postcondició:** L'usuari es redirigeix a la pantalla de login

Curs típic d'esdeveniments:	
Accions dels autors	Respostes del sistema
1. L'usuari accedeix al hamburger menú	
	2. El sistema mostrarà les opcions del menú
3. L'usuari selecciona tancar sessió	
	4. El sistema redirigeix a la pantalla de logueig

7.4.10. Valorar App

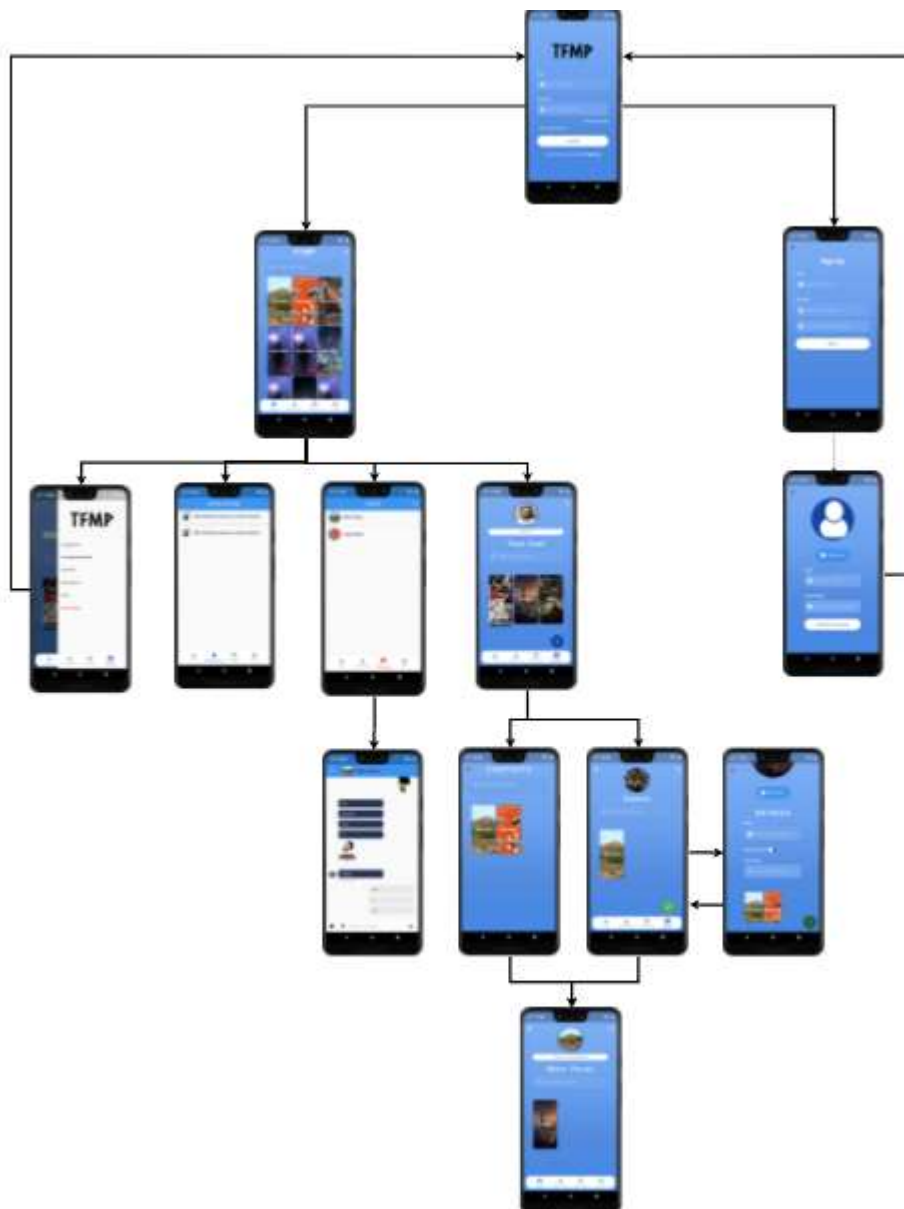
- **Propòsit:** L'usuari desitja valorar l'aplicació
- **Precondició:** L'usuari ha fet login en l'aplicació
- **Postcondició:** L'usuari es redirigit a l'aplicació en la Store del corresponent sistema operatiu

Curs típic d'esdeveniments:	
Accions dels autors	Respostes del sistema
1. L'usuari accedeix al hamburger menú	
	2. El sistema mostrà les opcions del menú
3. L'usuari selecciona valorar l'aplicació	
	4. El sistema redirigeix a la Store del dispositiu
5. L'usuari valora l'aplicació	

8. DISSENY DE L'APLICACIÓ

En aquest apartat s'explicarà de forma més detalla cada pantalla de la aplicació i amb un llenguatge més tècnic. També s'explicarà tot tipus de disseny que ha set necessari implementar en l'aplicació com seria la barra de navegació.

8.1. Mapa de navegació entre pantalles



IL·LUSTRACIÓ 15 - MAPA DE NAVEGACIÓ

8.2. Estil de navegació

L'estil que hem agafat es un estil clàssic i senzill per tal de que la gent el noti familiar. La barra de navegació, on es mostra en la primera imatge, estarà situada en la part baixa del dispositiu i contarà de quatre botons amb la seva icona i un Text amb el nom de la pantalla en qüestió. Per fer-ho més intuïtiu, com es pot apreciar en la imatge, un cop estàs en una d'aquestes pantalles, la icona i el text canviaran de color per indicar-te en quina pestanya de la navegació et trobes.



IL·LUSTRACIÓ 16 - BARRA DE NAVEGACIÓ

També tenim una altra barra de navegació que seria la AppBar. Aquesta barra es troba situada en la part superior de la pantalla. Aquesta barra és la que ens permet tornar a la pantalla anterior o obrir el submenú amb altres opcions, com seria tancar la sessió. Els botons estan situats seguint el estàndard de totes les aplicacions, en cas de voler anar a la pantalla anterior el boto es troba en la part superior esquerra amb la fletxa de retrosses. I el botó del menú sempre es troba en la part dreta superior, el qual sempre es desplega de dreta a esquerra.



IL·LUSTRACIÓ 17 - APPBAR

8.3. Disseny de pantalles

8.3.1. Login

Aquesta pantalla es mostrarà una vegada l'usuari obri l'aplicació per primera vegada o, en cas d'obrir sessió, no guardi les dades d'accés.

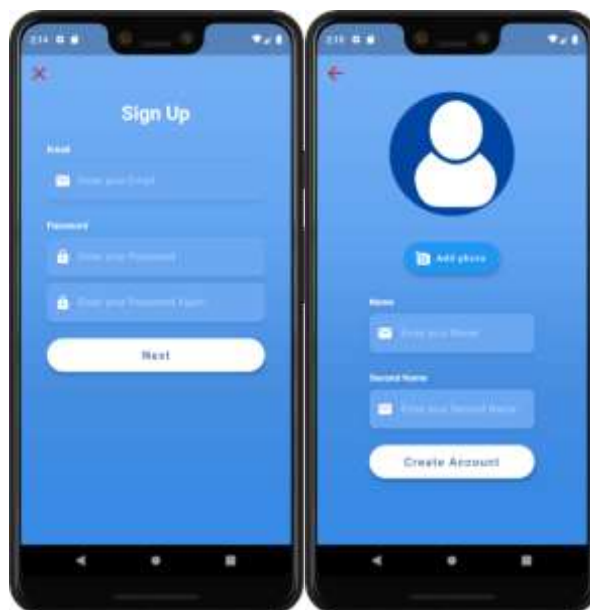
Aquesta pantalla esta dissenyada per a que primer de tot, a traves d'un Image.asset, mostri la imatge del logotip de l'aplicació al centre superior. A continuació de la imatge es mostrarà una unió d'un Text amb un TextField, el qual el Text ens indicarà el que tenim que introduir en el TextField, que en aquest cas el primer serà l'email. Seguidament trobarem la mateixa unió però per la contrasenya, i aquest TextField serà modificat per tal de que no es mostri el que s'està escrivint en ell. Sota aquest camps es troben un botó per restablir la contrasenya, en cas que l'usuari haguí oblidat la contrasenya del compte. A continuació, hi ha un checkbox per tal de voler recordar les dades de logueix i així cada cop que obrim l'aplicació no ens les demani. Seguidament es troba el botó de logueig i per últim, un botó el qual ens portarà a les pantalles de registre, que veurem en l'apartat següent.



IL·LUSTRACIÓ 18 - PANTALLA LOGIN

8.3.2. Registre

El registre es basa en dues pantalles, la primera pantalla és on s'ha d'introduir l'email i la contrasenya, dos cops per comprovar que no s'ha equivocat en ficar-la. Aquest camps tenen el mateix disseny e implementació que els de login. En cas que es deixi algun camp sense omplir, l'email ja ha sigut registrat anteriorment o que les dues contrasenyes no coincideixin, es mostrarà un missatge indicant l'error. Per últim, ens portarà a una altra pantalla on hauríem d'omplir les dades personals de l'usuari, la fotogràfica, el nom, etc. On el disseny és el mateix que la pantalla anterior però en aquest cas al principi tindrem un CircleAvatar per donar-li forma ovalada a la foto del perfil i dins d'aquest CircleAvatar ficarem un Image.asset per agafar la foto de perfil que l'usuari vulgui afegir del dispositiu. Sota aquest CircleAvatar es trobarà el FloatingActionButton, el qual executarà un codi per tal de agafar la foto de la galeria del dispositiu e introduir-la en el Image.asset



IL·LUSTRACIÓ 19 - PANTALLES DE REGISTRE

8.3.3. Pantalla d'inici

La pantalla d'inici ens mostrarà un GridView amb tots els usuaris de l'aplicació. Es mostrarà la foto de perfil, amb la implementació utilitzada anteriorment en l'apartat de registre, i el nom de l'usuari a través d'un TextView. En aquesta pantalla tindrem un buscador en la part superior per tal de si volem buscar algun usuari en concret, el qual utilitza un TextField per poder introduir la recerca. Si fem clic a un usuari ens portarà a la pantalla del perfil de l'usuari en qüestió, gràcies a que al GridView esta implementat el GestureDetector i el onTap(), el qual al fer clic sobre una casella del GridView podem saber de quina casella estem clicant.



IL·LUSTRACIÓ 20 - PANTALLA D'INICI

8.3.4. Vista perfil d'usuari

En el perfil d'usuari es mostrarà tota la informació bàsica de l'usuari. Primer es mostrarà la imatge de l'usuari seguidament d'un boto on, depenent de si és el teu perfil o no, mostrarà el boto per enviar sol·licitud/acceptar d'amistat, xatejar en cas de que siguin amics o els contactes, aquest últim es en cas que sigui el teu perfil d'usuari, com es mostra en la segona imatge. Per últim, trobem un buscador i tots els taulells que aquell usuari té. En el cercador pots introduir el nom d'algun taulell en concret per buscar-ho més fàcilment.



IL·LUSTRACIÓ 21 - PERFIL D'USUARI

8.3.5. Taulells

En aquestes pantalles es mostra tota la informació dels taulells. Primer de tot es mostra una imatge que ha set seleccionada per identificar el taulell. Seguidament es mostra el nom del taulell i per últim, tots els usuaris els quals l'usuari ha agrupat dins d'aquest taulell. També hi ha una barra de recerca, en cas que hi haguí molts usuaris dins del taulell poder trobar l'usuari en qüestió amb més facilitat. Com es pot apreciar en la primera imatge, a l'hora de crear un taulell és necessari ficar un nom i que aquest nom no sigui igual que cap taulell que aquell usuari haguí tingut. En cas que sigui un taulell privat els usuaris no el veuran en el perfil del usuari en qüestió.



IL·LUSTRACIÓ 22 - PANTALLA TAULELLS

8.3.6. Notificacions

En la pantalla de notificacions ens mostrarà en una llista on estaran els dos tipus de notificacions que un usuari pot rebre. Si l'usuari fa clic en una d'aquestes notificacions et portarà al perfil de l'usuari de la notificació en qüestió.

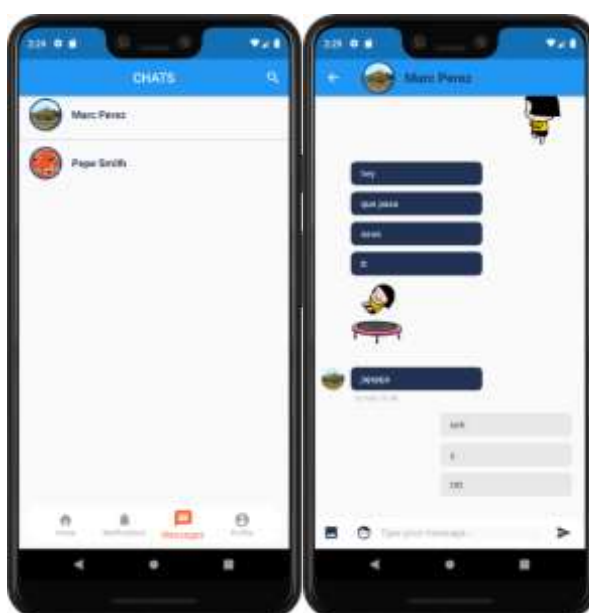
La implementació ha set una ListView, utilitzant un Divider per separar cada element de la llista amb una franja negra. Dintre del ListView retornem un ListTile que conté un Container, que dins d'ell es on hi ha la icona i el text. En el ListTile, s'implementa la funció de onTap(), per tal de saber si l'usuari ha clicat a la llista.



IL·LUSTRACIÓ 23 - PANTALLA NOTIFICACIONS

8.3.7. Xats

En la primera pantalla el que podem veure són totes les amistats que té l'usuari i amb les quals pot xatejar. Es mostrarà una ListView amb totes les amistats, amb l'implementació utilitzada en l'apartat anterior, i en la part superior hi ha una icona d'una lupa, la qual si fem clic sobre ella ens desplegarà un buscador per tal de cerca un usuari en concret i ser més fàcil trobar el xat amb aquell usuari. Un cop l'usuari elegeixi amb qui vol xatejar, t'enviarà a la segona pantalla la qual serà una pantalla de xat. Aquí es podrà veure tota la conversació que s'ha tingut amb aquest usuari en qüestió i es podrà enviar tant stickers, fotos o missatges de text.



IL·LUSTRACIÓ 24 - PANTALLA XATS

8.3.1. Contactes

En Contacte, el que podrem veure és un GridView amb totes les amistats que l'usuari té. Com es pot veure en anteriors pantalles, també hi ha un buscador en cas de voler buscar un contacte en concret. Un cop l'usuari faci clic a un contacte, l'aplicació et encaminarà a la pantalla del perfil de l'usuari en qüestió. La implementació d'aquesta pantalla és molt similar a la pantalla d'inici.



IL·LUSTRACIÓ 25 - PANTALLA CONTACTES

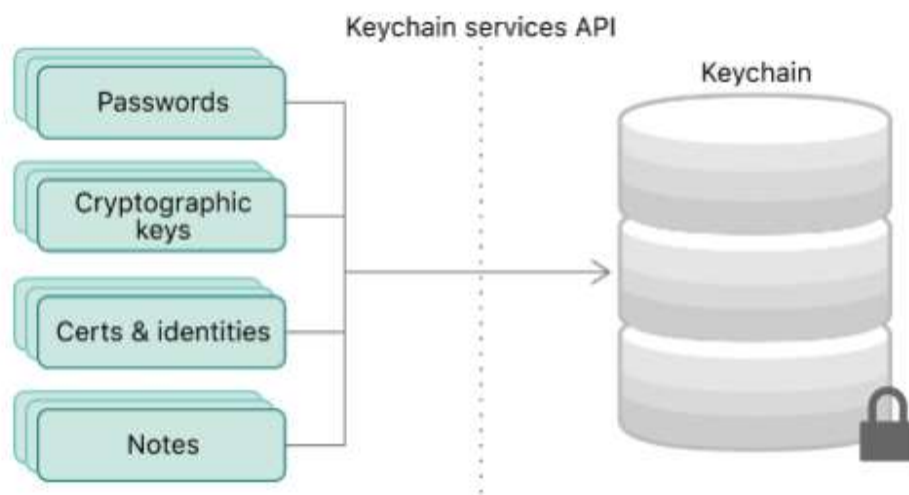
9. IMPLEMENTACIÓ

En aquest apartat a continuació podrem veure la implementació de les parts més importants de l'aplicació i el perquè s'ha decidit elegir dita implementació davant altres opcions.

9.1. Seguretat

A l'hora de guardar informació de l'usuari al terminal sempre s'utilitza Shared Preferences, que ens crea un fitxer al terminal amb una clau-valor i allí podem guardar tota mena d'informació de l'usuari, com per exemple alguna configuració de l'aplicació. Però això té un problema, ja que aquest fitxer no està xifrat i per tant no podem guardar informació sensible en aquest fitxer, com seria una contrasenya o dades bancàries. La solució ha set utilitzar el widget flutter_secure_storage, gràcies a què aquest widget utilitza Keychain si el dispositiu és iOS i en cas que sigui Android utilitza AES Encryption.

- **Keychain:** és un servei que ens ajuda a emmagatzemar petits bits de dades d'usuari en una petita base de dades encriptada anomenada Keychain. En aquesta base de dades podem guardar tota mena d'informació, com dades bancàries fins a petites notes. En la següent figura podem veure el funcionament d'aquest servei.



IL·LUSTRACIÓ 26 - ESTRUCTURA KEYCHAIN SERVICES

- **AES Encryption:** La clau secreta de AES es encriptada per RSA i aquesta clau es emmagatzemada al Keystore d'Android. Keystore et permet emmagatzemar claus criptogràfiques en un contenidor per a que resulti més difícil d'extreure del dispositiu. Keystore va ser introduït a Android 4.3 (API level 18), per tant no funciona per a versions anteriors.

Gràcies aquest widget utilitzarà, depenent del sistema operatiu del dispositiu, un mecanisme o un altre, i així tindrem la informació sensible més segura.

Per a la utilització d'aquest widget primer de tot el que haurem de fer és crear aquesta base de dades del widget, amb la següent comanda:

```
// Create storage  
final storage = new FlutterSecureStorage();
```

En cas de voler guardar alguna informació el que haurem de fer és utilitzar una clau-valor, per així a l'hora de voler buscar aquesta informació introduïm la clau ens retornarà el valor que hem guardat.

```
// Write value  
await storage.write(key: key, value: value);
```

```
// Read value  
String value = await storage.read(key: key);
```

En cas de voler sobre escriure la informació que té alguna clau, el que haurem de fer és tornar a escriure la clau. Però en cas que volem esborrar una clau i el seu valor haurem d'utilitzar la següent comanda.

```
// Delete value  
await storage.delete(key: key);
```

9.2. Connexió amb la Base de Dades

L'aplicació necessita una base de dades per a guardar totes les dades necessàries per al correcte funcionament com serien els taulells de cada usuari, els usuaris, etc.

Per resoldre aquesta necessitat podríem elegir:

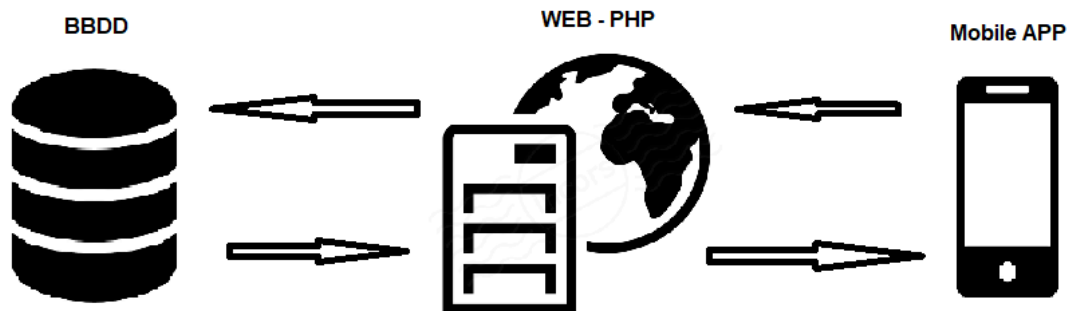
- **Firebase Realtime Database:** És una plataforma mòbil desenvolupada per Google en el 2014. Aquesta plataforma treballa a través del núvol, ho té tot guardat al núvol i utilitza un conjunt d'eines per a la creació i sincronització de projectes en temps real. El problema d'aquesta base de dades, que per a una aplicació de grans dimensions i per a una aplicació que té visió d'expansió aquesta base de dades es quedaria petita i a la llarga seria un problema.
- **WEB-PHP:** Un servei web el qual mantindrà tota la informació de la base de dades i a través de fitxer PHP pujats al servidor l'API executarà aquests fitxers els qual retornaran la informació pertinent de la sentència executada a la base de dades.
- **Widget mysql1:** Aquest widget ens proporciona una interfície fàcil per a utilitzar una base de dades MySQL. A través de Flutter, podem fer una connexió directa amb la base de dades i realitzar les consultes necessàries sense necessitat de tercers. A continuació podem veure una línia de codi on realitzem un insert directament a la base de dades i podem veure que es molt senzill e intuïtiu.

```
var result = await conn.query('insert into users (name, email, age) values (?, ?, ?)',  
['Bob', 'bob@bob.com', 25]);
```

El problema d'aquest widget és que no és un widget verificat per l'equip de flutter, per tant es pot quedar obsolet. També al provar aquest widget hem vist que dóna molts problemes per a la connexió amb la base de dades.

Finalment, avaluant aquest possibilitat s'ha triat fer una connexió WEB-PHP, ja que es una connexió molt bàsica, fàcil d'implementar i funciona a la perfecció.

Per fer aquesta connexió l'aplicació es connecta a un servei web el qual té emmagatzemats fitxers php que el que fan es cridar a la base de dades i en cas de que sigui una crida select, el que farà es retornar la resposta de la base de dades. Per tant el servei WEB-PHP, com es pot veure en la figura següent, fa de pont entre l'aplicació mòbil i la base de dades.



IL·LUSTRACIÓ 27 - ESTRUCTURA DEL SISTEMA DE CONNEXIÓ

Com podem veure a la imatge següent el que fem és cridar al fitxer que tenim emmagatzemat a la web a través de la ruta URL i en el cas que aquest fitxer PHP rebi alguna variable s'haurà que passar pel body.

```

_checkUserExist() async {
  final response = await http.post("https://www.martabataalla.com/flutter/wenect/loginUser.php",
    body: {
      "email": _emailController.text,
      "password": _passwordCoded[0]
    });

  var dataUser = json.decode(response.body);

  if(dataUser.length>0){
    // Login Correct
  } else {
    // Login Error
  }
}

```

IL·LUSTRACIÓ 28 - EXEMPLE DE CODI PER A CONNEXIÓ PHP

Aquesta funció el que farà es esperar una resposta del servidor i la descodificarà, en cas que haguí rebut una resposta voldrà dir que el select ens ha retornat l'usuari, per tant l'usuari existeix.

En la imatge següent podem veure com seria el fitxer php que realitza la connexió a la base de dades i executa la consulta. Primer de tot executarà un nou fitxer php, que és el que realitza la connexió amb la base de dades. També podem veure que aquest fitxer té que rebre dos variables, que serien l'email i la contrasenya via POST. Seguidament, el que fa es llançar la sentència que li passem a la base de dades i guardar la resposta en una variable. Per últim, ja que és un select i pot retorna més de una fila, es va guardant en un array i finalment aquest array es codifica via json per enviar-lo a l'aplicació via echo.

```
<?php

    include 'conexion.php';

    $email = $_POST['email'];
    $password = $_POST['password'];

    $queryResult=$connect->query("SELECT *
    FROM `Users`
    WHERE user_email='".$email."' AND user_password='".$password."'");

    $result=array();

    while($fetchData=$queryResult->fetch_assoc()){
        $result[]=$fetchData;
    }

    echo json_encode($result);
?>
```

IL·LUSTRACIÓ 29 - EXEMPLE FITXER PHP

Això seria un exemple de com funcionaria la connexió de l'aplicació amb la base de dades a través de un servei WEB-PHP.

9.3. Internalització

En l'aplicació ha set implementada la internalització, que ens farà tindre un target de gent molt més ampli que si l'aplicació sols tingués un idioma. En aquest cas l'idioma principal de l'aplicació ha set el anglès, ja que de base ja es podria utilitzar en qualsevol país.

Primer de tot per a un correcte funcionament haurem de veure en quin idioma està el dispositiu, ja que en cas que el dispositiu estigui en un idioma que l'aplicació té traducció haurem de carregar aquell idioma, si el dispositiu està en un idioma que l'aplicació no té traducció, haurem de deixar l'anglès com idioma predeterminat.

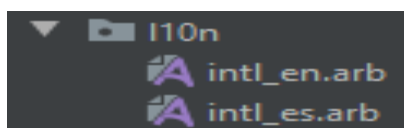
```
Locale myLocale = Localizations.localeOf(context);
```

Amb aquesta línia de comanda sabrem en quin idioma està el dispositiu.

Flutter, ens proporciona un paquet muti-idioma anomenat `flutter_localizations` que ens proporciona implementacions multi-idioma de les interfícies de localització, fent-ho tot molt més fàcil d'implementar. Sols cal implementar les següents línies de codi i l'aplicació ja sabrà quin arxiu tindrà que carregar dependent de l'idioma del dispositiu.

```
localizationsDelegates: [  
  GlobalMaterialLocalizations.delegate,  
  GlobalWidgetsLocalizations.delegate,  
  GlobalCupertinoLocalizations.delegate,  
  S.delegate,  
],  
supportedLocales: S.delegate.supportedLocales,
```

Un cop tenim tot ja tot configurat el que hem de fer és crear el fitxer on estaran tots els textos de l'aplicació que tenen traducció. Aquests textos estaran acompanyats d'una clau i valor per així poder identificar-los. La nomenclatura dels fitxers té que ser sempre la mateixa, ja que és la nomenclatura que el paquet `flutter_localizations` té implementada, i és la següent:



IL·LUSTRACIÓ 30 - EXEMPLE FITXERS INTERNALITZACIÓ NECESSARIS

Es crea una carpeta que es dirà l10n i allí guardarem tots els fitxers d'idioma que volem que tingui la nostra aplicació. En aquest cas podem veure que té amb anglès i espanyol. Dins d'aquest fitxer ficarem les claus i els textos que volem traduir.

En la imatge següent es pot veure la clau-valor per als textos en espanyol.

```
{  
  "preferencesText": "Preferencias de usuario",  
  "selImageText": "Seleccionar imagen"  
}
```

IL·LUSTRACIÓ 31 - EXEMPLE INTERNALITZACIÓ DEL TEXT AMB ESPANYOL

I en aquesta última imatge es pot veure que les claus són les mateixes per a anglès i espanyol, però el que canvia es el text, que en aquest cas està en anglès. Això és degut a que on aniria aquets text el que es ficarà es la clau del text i el widget dependent del idioma del dispositiu carregarà un fitxer o un l'altre.

```
{  
  "preferencesText": "Shared preferences",  
  "selImageText": "Select an image"  
}
```

IL·LUSTRACIÓ 32 - EXEMPLE INTERNALITZACIÓ DEL TEXT AMB ANGLES

9.4. Barra de navegació

Per a qualsevol aplicació és necessari tindre una barra de navegació situada sota la pantalla per tal que sigui molt més fàcil per a l'usuari navegar per l'aplicació. En aquesta barra es fiquen les funcionalitats més utilitzades i principal de l'aplicació perquè l'usuari tingui un accés fàcil a elles i no hagi de fer més clics dels necessaris per accedir a aquesta pantalla.

Per fer aquesta barra de navegació hem utilitzat un widget anomenat `persistent_bottom_nav_bar`. No hem utilitzat la classe que flutter ens proporciona que seria la de `BottomNavigationBarClass`, ja que dona bastants problemes en el tema de la navegació entre les pantalles, el boto de tornar a la pantalla anterior i en cas de no voler la barra de navegació en algunes pantalles és bastant complicat de realitzar, per això s'ha agafat aquest widget que s'ha comentat anteriorment.

Aquest widget el que ens proporciona és:

- Barra de navegació inferior persistent altament personalitzable.
- Possibilitat d'impulsar pantalles noves amb o sense barra de navegació inferior.
- 20 estils per a la barra de navegació inferior.
- Inclou funcions per empènyer la pantalla amb o sense la barra de navegació inferior, és a dir, `pushNewScreen()` i `pushNewScreenWithRouteSettings()`.
- Basat en la barra de navegació inferior de Cupertino (iOS) de Flutter.
- Pot ser translúcid per a una pestanya concreta.
- Estil personalitzat per a la barra de navegació.
- Gestiona el botó de retrocés d'Android de maquinari / programari

A continuació es pot veure alguns dels dissenys predeterminats que el widget et proporciona.



IL·LUSTRACIÓ 33 - TIPUS DE DISSENYS DE LA BARRA DE NAVEGACIÓ

Per a utilitzar aquest widget primer de tot l'hauréu de configurar i això es fa directament a la classe principal del projecte, que seria MyApp. A continuació, es veu un exemple de creació d'aquesta barra de navegació.

```
class MyApp extends StatelessWidget {
  const MyApp({Key key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return PersistentTabView(
      context,
      controller: _controller,
      screens: _buildScreens(),
      items: _navBarsItems(),
      confineInSafeArea: true,
      backgroundColor: Colors.white,
      handleAndroidBackPressed: true,
      resizeModeAvoidBottomInset: true, // This needs to be true if you want to move up
      stateManagement: true,
      hideNavigationBarWhenKeyboardShows: true, // Recommended to set 'resizeToAvoidBottomInset'
      decoration: NavBarDecoration(
        borderRadius: BorderRadius.circular(10.0),
        colorBehindNavBar: Colors.white,
      ),
      popAllScreensOnTapOfSelectedTab: true,
      popActionScreens: PopActionScreensType.all,
      itemAnimationProperties: ItemAnimationProperties( // Navigation Bar's items ani
        duration: Duration(milliseconds: 200),
        curve: Curves.ease,
      ),
      screenTransitionAnimation: ScreenTransitionAnimation( // Screen transition ani
        animateTabTransition: true,
        curve: Curves.ease,
        duration: Duration(milliseconds: 200),
      ),
      navBarStyle: NavBarStyle.style1, // Choose the nav bar style with this property
    );
  }
}
```

IL·LUSTRACIÓ 34 - EXEMPLE CODI CREACIÓ BARRA DE NAVEGACIÓ

Com es pot veure en l'exemple anterior, en l'opció de screens, s'haurà de crear una llista de widgets el qual ens retornarà totes les pantalles que la barra de navegació tindrà. Com es pot veure en l'exemple següent tindrà tres pantalles.

```
List<Widget> _buildScreens() {
  return [
    PersonalWallScreen(),
    MainWallScreen(),
    CreateBoardScreen(),
  ];
}
```

IL·LUSTRACIÓ 35 - EXEMPLE CODI CONFIGURACIÓ BARRA DE NAVEGACIÓ

També en l'opció d'items haurem de tindre una llista la qual aquesta ens retornarà un `PersistentBottomNavBarItem`, que seran els botons que apareixeran en aquesta barra de navegació. Aquest elements tindran una icona, el text que vols que aparegui del boto i els colors que voldràs que tinguin en cas de tindre seleccionat el boto o no.

```

List<PersistentBottomNavBarItem> _navBarsItems() {
  return [
    PersistentBottomNavBarItem(
      icon: Icon(Icons.home),
      title: "Home",
      activeColor: Colors.blue,
      inactiveColor: Colors.grey,
    ), // PersistentBottomNavBarItem
    PersistentBottomNavBarItem(
      icon: Icon(Icons.search),
      title: ("Search"),
      activeColor: Colors.teal,
      inactiveColor: Colors.grey,
    ), // PersistentBottomNavBarItem
    PersistentBottomNavBarItem(
      icon: Icon(Icons.add),
      title: ("Add"),
      activeColor: Colors.blueAccent,
      inactiveColor: Colors.grey,
      activeContentColor: Colors.white,
    ), // PersistentBottomNavBarItem
  ];
}
    
```

IL·LUSTRACIÓ 36 - EXEMPLE CODI CONFIGURACIÓ BARRA DE NAVEGACIÓ 2

Per últim, en l'opció de `navBarStyle`, li indicarem quin tipus de disseny volem que tingui la barra de navegació de tots els que et proporciona el widget.

Un cop tenim configurat ja el widget, per a navegar entre pantalles el que es farà és utilitzar la funció `pushNewScreen()`. En aquesta funció el que li passarem es el context de la pantalla, a quina pantalla es vol navegar i si volem que en la pantalla que naveguem, tenir la barra de navegació o no.

```
pushNewScreen(
  context,
  screen: MainScreen(),
  withNavBar: true, // OPTIONAL VALUE. True by default.
  pageTransitionAnimation: PageTransitionAnimation.cupertino,
);
```

IL·LUSTRACIÓ 37 - EXEMPLE CODI NAVEGACIÓ A UNA ALTRA PANTALLA

9.5. Missatgeria

En l'aplicació hem implementat un sistema de missatgeria entre usuaris que són amics. Per tal que aquests usuaris es puguin comunicar per l'aplicació.

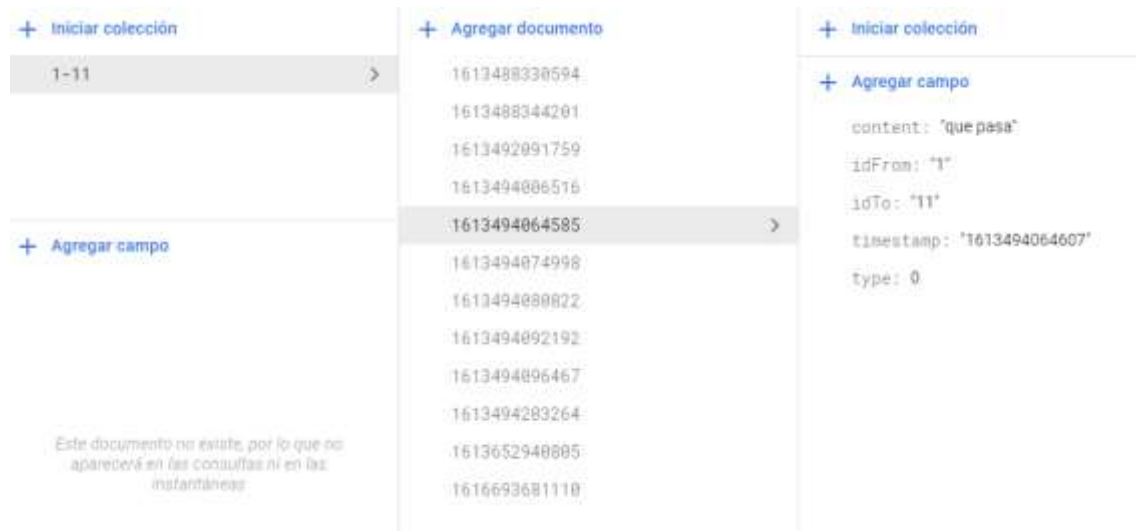
Per implementar aquest sistema hem optat per utilitzar l'eina Firebase, comentada anteriorment. Hem optat per aquesta opció com que gràcies al seu funcionament és molt més senzill implementar aquest sistema de missatgeria, com que t'ho agrupa tot per col·leccions i documents, i això fa tot molt més senzill i menys caòtic que aplicar-ho a la base de dades MySQL on es guarden totes les dades dels taulells dels usuaris. També aquesta base de dades ens proporciona una sincronització instantània que ens ajudarà al fet que no es perdi cap missatge enviat i no s'hagi d'implementar cap mena de refrec perquè els usuaris rebin els missatges a temps real.

L'estructura d'aquesta base de dades és la següent:



IL·LUSTRACIÓ 38 - ESTRUCTURA FIREBASE

Tenim una col·lecció anomenada “messages”. Aquesta col·lecció es on es guardaran tots els missatges de l’aplicació. Seguidament tenim un document, que ens ajudarà a saber els usuaris els quals estan sent enviats els missatges, bàsicament sabrem de quin xat és aquest missatge. L’identificador dels documents serà el ID dels usuaris implicats en aquest xat. Com es pot veure en la imatge en aquest cas serà l’usuari amb el ID 1 i l’11.



IL·LUSTRACIÓ 39 - ESTRUCTURA FIREBASE 2

Seguidament tindrem una nova col·lecció amb aquest mateix identificador on dins d’aquesta col·lecció guardarem un document el qual aquest document portarà tota la informació del missatge que ha set enviat. Com podem veure en la imatge aquests documents tindran uns camps específics on guardaran tota la informació. Aquests camps son:

- **Content:** El missatge que ha set enviat
- **idFrom:** L’ID del usuari del qual ve el missatge
- **idTo:** L’ID del usuari al qual va el missatge
- **timestamp:** L’hora la qual s’ha enviat aquest missatge
- **type:** El tipus de missatge (0 = text, 1 = imatge, 2 = sticker)

L’aplicació té una pantalla on sortiran tots els amics que tens els quals pots xatejar i un cop selecciones a un amic carregarà la col·lecció amb els ID’s dels dos usuaris amb tots els missatges que s’han enviat.

10. CONCLUSIONS

Després d'acabar el projecte puc dir que, en l'àmbit personal, he après molt i he millorant molt en el tema de desenvolupament de aplicacions multi-plataforma i en la creació d'elles des de zero.

Ha estat un treball satisfactori, ja que de base partia amb molta incertesa de si podria realitzar el treball amb èxit, ja que la tecnologia utilitzada era completament nova per a mi i no havia vist molt sobre ella, afegint que en d'altres aspectes de programació, com es PHP, feia anys que no desenvolupava res.

Conseqüència de que feia anys que no programava amb PHP, hi han hagut aspectes que no s'han pogut realitzar com es tenien planejats d'un inici, però s'han anat trobant solucions per tal de completar el funcionament que volíem. Això també ha passat amb el tema de Flutter, ja que al no saber res de base, sobre el desenvolupament amb Flutter, hi han hagut aspectes que volíem implementar que ens han costat més que altres que no s'assemblaven amb el desenvolupament Android, però finalment s'han anat pogut trobant solucions i finalment al veure el resultat és molt més gratificant al treure endavant un projecte partint de zero.

11. TREBALLS FUTURS

Degut a que el projecte ha estat adaptat per tal de poder ser implementat en el termini que té un treball de fi de grau han quedat coses en el tinter les quals en un futur es podrien implementar, per tal de que l'aplicació sigui molt més funcional i extensa.

Aquestes funcionalitats serien:

- 1- Creació de taulells per invitació. Aquest taulells serien grups tancats on tots els usuaris tindrien aquest taulell.
- 2- Implementar xat pels taulells amb invitació, per tal de poder discutir d'alguns temes relacionats amb el taulell.
- 3- Implementar una recerca d'amistats més ajustada depenen de les teves amistats. Crear una recerca on es basa en les amistats que tens i les amistats dels teus contactes, per tal de trobar persones més properes i de confiança.
- 4- Mostrar taulells més visitats de les amistats en la pantalla principal.
- 5- Proba de stres amb molts usuaris, tant per la base de dades com l'API.

12. AGRAÏMENTS

Primer de tot voldria donar les gràcies al professor Jordi Agost, ja que jo sempre tenia pensat fer un projecte de desenvolupament mòbil i ell va confiar en mi per fer un projecte, que a priori, era un projecte bastant difícil, ja que utilitzava nous conceptes i també conceptes que no els hem treballat molt durant el grau.

Treballar amb ell ha set molt senzill i molt enriquidor, ja que té un ampli coneixement de les tecnologies i sempre esta actualitzant-se de tot el que es diu i surt com a novetat.

Per últim, voldria donar les gràcies a la professora Montserrat Sendin, per confiar en nosaltres en un projecte que no era especialment sobre el desenvolupament amb Android, que és el que principalment es realitza en el grau, però que a la fi es tracta de desenvolupar una aplicació mòbil.

Sempre ens ha ajudat en tot el que li anàvem demanant per tal de poder realitzar el projecte correctament, per tant volia donar-li les gracies ja que sense ella tampoc hauria set possible realitzar-ho.

13. WEBGRAFIA

[1] What is Mobile Applications Development? Disponible en:

<https://aws.amazon.com/es/mobile/mobile-application-development/>

[2] Swift. Disponible en:

<https://developer.apple.com/swift/>

[3] Android vs iOS Market Share 2021. Disponible en:

<https://leftronic.com/blog/android-vs-ios-market-share/#:~:text=Google%20Android%20and%20Apple%20iOS,share%20in%202019%20is%2022.17%25.>

[4] Kotlin. Disponible en:

<https://developer.android.com/kotlin>

[5] Ventajas y desventajas de programar en JAVA. Disponible en:

<https://adictoalcodigo.blogspot.com/2016/07/ventajas-y-desventajas-de-programar-en.html>

[6] What is the difference between a Mobile App and a Web APP?. Disponible en:

<https://careerfoundry.com/en/blog/web-development/what-is-the-difference-between-a-mobile-app-and-a-web-app/#:~:text=Snapchat%2C%20Instagram%2C%20Google%20Maps%20and,to%20be%20downloaded%20or%20installed.>

[7] 21 of the Best Examples of Mobile Website Design. Disponible en:

<https://blog.hubspot.com/marketing/mobile-website-design-examples>

[8] Flutter visto con las gafas de programador web. Disponible en:

<https://www.paradigmadigital.com/dev/flutter-visto-con-gafas-programador-web/>

[9] Flutter and Fuchsia – The next big thing. Disponible en:

<https://medium.com/flutterdevs/flutter-and-fuchsia-the-next-big-thing-b0688674b681>

[10] Introduction to widgets - Flutter. Disponible en:

<https://flutter.dev/docs/development/ui/widgets-intro>

[11] Android vs iOS Market Share 2021. Disponible en:

https://www.tutorialspoint.com/flutter/flutter_architecture_application.htm#:~:text=The%20core%20concept%20of%20the,application%20is%20itself%20a%20widget.

[12] Flutter architectural overview. Disponible en:

<https://flutter.dev/docs/resources/architectural-overview>

[13] Flutter App architecture. Disponible en:

<https://medium.com/swlh/flutter-app-architecture-1a2a2a4124c8>

[14] Fuchsia OS y Flutter. Disponible en:

<https://medium.com/comunidad-flutter/fuchsia-os-y-flutter-e1c9e290c887>

[15] Flutter every where - BBVA. Disponible en:

<https://www.bbvanexttechnologies.com/flutter-everywhere/>

[16] Top Apps made with Flutter. Disponible en:

<https://www.thedroidsonroids.com/blog/apps-made-with-flutter#:~:text=Xianyu%2C%20Alibaba&text=Alibaba%2C%20the%20world's%20biggest%20online,use%20this%20app%20every%20day.>

[17] Flutter multiplataforma. Disponible en:

<https://platzi.com/blog/flutter-multiplataforma/>

[18] Set up an editor - Flutter. Disponible en:

<https://flutter.dev/docs/get-started/editor?tab=androidstudio>

[19] Firebase Realtime Database. Disponible en:

<https://firebase.google.com/docs/database?hl=es-419>

[20] Widget mysql1 - Flutter. Disponible en:

<https://pub.dev/packages/mysql1>

[21] ¿Qué es PHP?. Disponible en:

https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=492:ique-es-php-y-ipara-que-sirve-un-potente-lenguaje-de-programacion-para-crear-paginas-web-cu00803b&catid=70&Itemid=193

[22] Internacionalizando aplicaciones Flutter. Disponible en:

<https://esflutter.dev/docs/development/accessibility-and-localization/internationalization>

[23] Flutter fácil de internacionalizar. Disponible en:

<https://medium.com/@danelop.ecu/flutter-f%C3%A1cil-de-internacionalizar-usando-provider-y-json-320ffe7b3fbd>

[24] Widget persistent bottom navigation bar - Flutter. Disponible en:

https://pub.dev/packages/persistent_bottom_nav_bar

[24] Widget bottom navigation bar class- Flutter. Disponible en:

<https://api.flutter.dev/flutter/material/BottomNavigationBar-class.html>

[25] Flutter 2.2 – Whats new. Disponible en:

<https://flutter.dev/docs/whats-new>